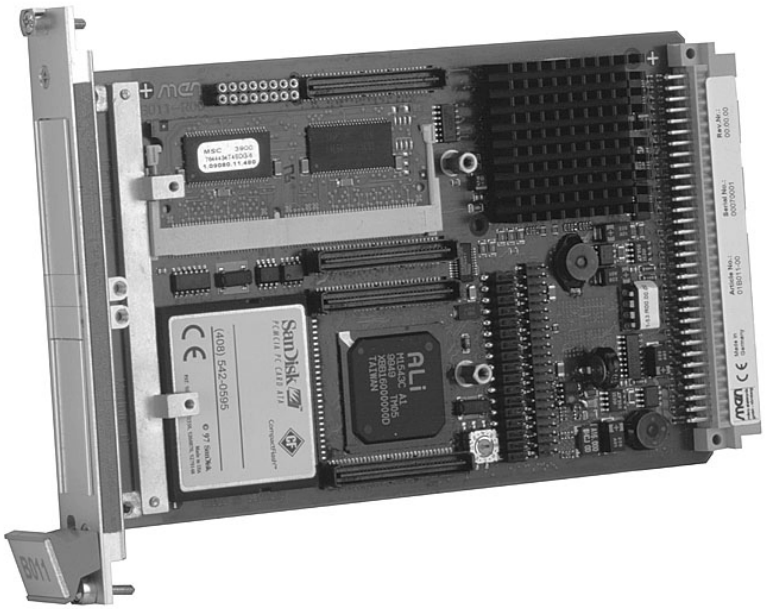
Embedded Solutions

B11 – 3U VMEbus PowerPC SBC



User Manual

Board-Level Computers for Industrial Applications

men mikro elektronik gmbh • nürnberg

B11 – 3U VMEbus PowerPC SBC

The B11 is a high-speed PowerPC based VMEbus card with complete single-board computing functionality. It is optimized for industrial requirements in control and instrumentation in terms of functionality, environmental conditions and cost. The B11 is especially prepared for all types of industrial qualification such as extended temperature range (-40..+85°C), shock, vibration, humidity etc.

The computing core of the B11 is the powerful MPC8245 Kahlua II processor with its internal PowerPC 603 and 300MHz clock frequency. The B11 is equipped with one SO-DIMM SDRAM slot, a CompactFlash slot and soldered boot Flash. This guarantees maximum flexibility in memory configuration. The B11 hosts two serial interfaces, IDE hard-disk controller, keyboard, mouse and USB as well as a real-time clock and watchdog. Two local PC•MIP mezzanine slots are available for flexible and individual workstation I/O extensions such as Ethernet, graphics, SCSI, additional serial lines, field busses etc.

The non-transparent PCI-to-VME bridge is a very fast and cost-competitive in-house design. Using a dual-ported SRAM, the B11 speeds up to a transfer rate of 7MB/s for master single cycle operations and up to 13MB/s for slave burst operations.

Despite the full functionality of the B11 and even with 2 PC•MIP slots, the complete CPU card needs only one 3U slot in a VMEbus system.

Technical Data

CPU

- Motorola PowerPC
  - MPC8245
  - 300MHz

Memory

- Level 1 Cache integrated in MPC8245
  - 16K data cache
  - 16K instruction cache
- SDRAM SO-DIMM up to 256MB
  - 64 bits
- Flash 2MB
  - 8 bits
- Serial EEPROM 2KB
  - For factory settings
- CompactFlash (TM) interface (true IDE)

Interfaces

- Two serial COM ports
  - Physical interface using SA adapter via 10-pin ribbon cable
  - RS232..RS485, isolated or not: for free use in system (e.g. cable to front)
- IDE
- Keyboard/mouse PS2
- USB

Local PCI Bus

- PCI Spec. 2.1 compliant
- 32 bit data bus, 33MHz, 3.3V

Local Extensions

- PC•MIP I/O at front panel
  - Two PC•MIP mezzanine extension slots
  - Compliant with PC•MIP specification (Type I/II slots)

VMEbus

- Slot-1 function with autodetection
- Master D08(EO):D16:A24:A16:RMW; transfer rate max. 7MB/s
- Slave D08(EO):D16:A24:BLT; transfer rate max. 30MB/s
- Interrupter D08(O):I(7-1):ROAK
- Interrupt handler D08(O):IH(7-1)
- Single level 3 fair requester
- Single level 3 arbiter
- Bus timer
- Arbitration timer
- Utility functions

Miscellaneous

- Real-time clock
- Watchdog and hardware monitor for on-board temperature control
- Hex switch for user settings

Electrical Specifications

- Supply voltage/power consumption:
  - +5V (4.75V..5.25V), 6W
- MTBF: 124,000h @ 50°C

Mechanical Specifications

- Dimensions: standard single Eurocard, 100mm x 160mm
- Weight: 138g

Environmental Specifications

- Temperature range (operation):
  - 0..+60°C or -40..+85°C
  - Airflow: min. 10m³/h
- Temperature range (storage): -40..+85°C
- Relative humidity range (operation): max. 95% non-condensing
- Relative humidity range (storage): max. 95% non-condensing
- Altitude: -300m to + 3,000m
- Shock: 15g/0.33ms, 6g/6ms
- Vibration: 1g/5..2,000Hz

Safety

- PCB manufactured with a flammability rating of 94V-0 by UL recognized manufacturers

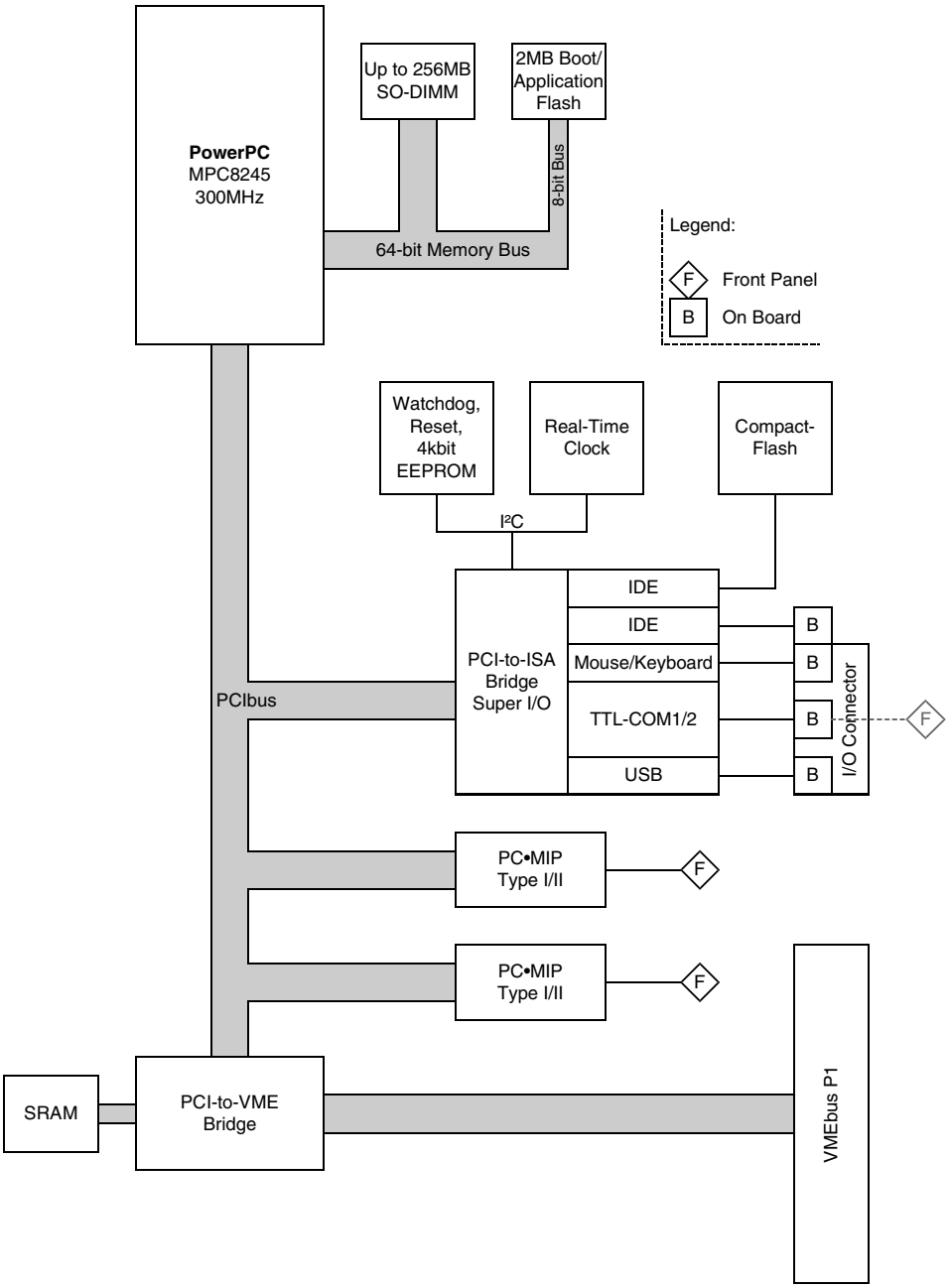
EMC

- Tested according to EN 55022 / 1999-05 (radio disturbance) and EN 55024 / 1999-05 (immunity) with regard to CE conformity

Software

- MENMON
- VxWorks
- Linux
- OS-9

Block Diagram



Product Safety

Fuses

This board contains fuses. If you need to replace a fuse, make sure you adhere to the following types and ratings:

Component	Current Rating	Type	Size
S1	1.5A	Fast	1206
S2	3A	Fast	1206

For component locations, see Chapter 5.3 Component Plans on page 84.

Electrostatic Discharge (ESD)

Computer boards and components contain electrostatic sensitive devices. Electrostatic discharge (ESD) can damage components. To protect the board and other components against damage from static electricity, you should follow some precautions whenever you work on your computer.

- Power down and unplug your computer system when working on the inside.
- Hold components by the edges and try not to touch the IC chips, leads, or circuitry.
- Use a grounded wrist strap before handling computer components.
- Place components on a grounded antistatic pad or on the bag that came with the component whenever the components are separated from the system.
- Store the board only in its original ESD-protected packaging. Retain the original packaging in case you need to return the board to MEN for repair.

About this Document

This user manual describes the hardware functions of the board, connection of peripheral devices and integration into a system. It also provides additional information for special applications and configurations of the board.

The manual does not include detailed information on individual components (data sheets etc.). A list of literature is given in the appendix.

History

Edition	Description	Technical Content	Date of Issue
E1	First edition	Holger Schubert, Thomas Eckert, Klaus Popp	2001-04-27
E2	Second edition	Holger Schubert, Thomas Eckert, Klaus Popp	2002-11-07
E3	Third edition	Holger Schubert, Thomas Eckert, Klaus Popp	2003-01-09

Conventions

This sign marks important notes or warnings concerning proper functionality of the product described in this document. You should read them in any case.

italics — Folder and file names are printed in *italics*.

bold — **Bold** type is used for emphasis.

hyperlink — Hyperlinks are printed in blue color.

The globe will show you where hyperlinks lead directly to the Internet, so you can look for the latest information online.

`0xFF` — Hexadecimal numbers are preceded by "0x", which is the usual C-language convention, and are printed in a monospace type, e.g. `0x00FFFF`.

IRQ# /IRQ — Signal names followed by "#" or preceded by a slash ("/") indicate that this signal is either active low or that it becomes active at a falling edge.

in/out — Signal directions in signal mnemonics tables generally refer to the corresponding board or component, "in" meaning "to the board or component", "out" meaning "coming from it".

Vertical lines on the outer margin signal technical changes to the previous edition of the document.

Copyright Information

MEN reserves the right to make changes without further notice to any products herein. MEN makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does MEN assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages.
"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts.
MEN does not convey any license under its patent rights nor the rights of others.
MEN products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the MEN product could create a situation where personal injury or death may occur. Should Buyer purchase or use MEN products for any such unintended or unauthorized application, Buyer shall indemnify and hold MEN and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that MEN was negligent regarding the design or manufacture of the part.

All brand or product names are trademarks or registered trademarks of their respective holders.

Information in this document has been carefully checked and is believed to be accurate as of the date of publication; however, no responsibility is assumed for inaccuracies. MEN will not be liable for any consequential or incidental damages arising from reliance on the accuracy of this document. The information contained herein is subject to change without notice.

Copyright © 2003 MEN Mikro Elektronik GmbH. All rights reserved.

Please recycle

Germany
MEN Mikro Elektronik GmbH
Neuwieder Straße 7
90411 Nuremberg
Phone +49-911-99 33 5-0
Fax +49-911-99 33 5-99
E-mail info@men.de
www.men.de

France
MEN Mikro Elektronik SA
18, rue René Cassin
ZA de la Châtelaine
74240 Gaillard
Phone +33 (0) 450-955-312
Fax +33 (0) 450-955-211
E-mail info@men-france.fr
www.men-france.fr

UK
MEN Micro Ltd
Whitehall, 75 School Lane
Hartford, Northwich
Cheshire UK, CW8 1PF
Phone +44 (0) 1477-549-185
Fax +44 (0) 1477-549-178
E-mail info@menmicro.co.uk
www.menmicro.co.uk

USA
MEN Micro, Inc.
3740 North Josey Lane, Suite 203
Carrollton, TX 75007
Phone 972-939-2675
Fax 972-939-0055
E-mail sales@menmicro.com
www.menmicro.com

Contents

1 Getting Started 14
1.1 Map of the Board 14
1.2 Configuring the Hardware 15
1.3 Integrating the Board into a System 15
1.4 Installing Operating System Software 16

2 Functional Description 17
2.1 Power Supply 17
2.2 Clock Supply 17
2.3 PowerPC CPU 17
2.3.1 General 17
2.3.2 Heat Sink 17
2.4 Bus Structure 18
2.4.1 Host-to-PCI Bridge 18
2.4.2 Local PCI Bus 18
2.4.3 PCI-to-ISA Bridge 18
2.4.4 PCI-to-VMEbus Bridge 18
2.5 Memory 19
2.5.1 SDRAM 19
2.5.2 Flash 20
2.5.3 CompactFlash 21
2.5.4 EEPROM 22
2.6 PC•MIP Slots 23
2.6.1 Installing PC•MIPs 23
2.6.2 PC•MIP Connectors 24
2.7 IDE Interface 25
2.8 I/O Connector 27
2.8.1 Serial Ports COM1/COM2 29
2.9 Hardware Monitor 33
2.10 Real-Time Clock and NVRAM 33
2.11 Watchdog 33
2.12 User-Defined Hex Switch 34
2.13 VMEbus Interface 35
2.13.1 SRAM 35
2.13.2 Slot-1 Function 35
2.13.3 Interrupter 35
2.13.4 Interrupt Handler 35
2.13.5 Bus Errors 36
2.13.6 Master Access 36
2.13.7 Atomic Operations 36
2.13.8 PCI Configuration Space Registers 37
2.13.9 Runtime Registers 38
2.13.10 Connection 41
2.14 IEEE 1149.1 (JTAG)/COP Test Interface 42

3 MENMON 43
3.1 General 43
3.2 Console 43
3.3 B11 MENMON Memory Map 44
3.4 MENMON Start-up 45
3.4.1 User LEDs 45
3.4.2 Boot Sequence 45
3.4.3 Configuring the MENMON Start-up Procedure 45
3.4.4 Self Tests 46
3.5 MENMON Boot Methods for Client Programs 48
3.5.1 MENMON BIOS Devices 48
3.5.2 Disk Boot 50
3.5.3 Network Boot 55
3.6 Updating Flash Devices 57
3.6.1 Download via Serial Interface 57
3.6.2 Performing the Download 58
3.6.3 Update from Disk or Network 59
3.7 MENMON User Interface 60
3.7.1 Command Line Editing 60
3.7.2 Numerical Arguments 60
3.7.3 MENMON Command Overview 61
3.8 Board Setup 62
3.8.1 ALI 1543 62
3.8.2 PCI Auto-Configuration 62
3.8.3 SDRAM DIMM Configuration 64
3.8.4 VMEbus Configuration 64
3.8.5 Watchdog Configuration 65
3.8.6 Hex Switch 65
3.9 MENMON System Calls 66
3.9.1 Invoking System Calls 66
3.9.2 System Calls 67
3.10 VxWorks Bootline 72
3.10.1 Additional MENMON Parameters 73

4 Organization of the Board 74
4.1 Memory Mappings 74
4.1.1 Processor View of the Memory Map 74
4.1.2 PCI/ISA I/O Space Memory Map 76
4.1.3 VMEbus Memory Mapping 77
4.2 Interrupt Handling 78
4.2.1 Nonmaskable Interrupts 79
4.2.2 Maskable Interrupts 79
4.3 Implementation of M1543 PCI-to-ISA Bridge 80
4.4 SMB Devices 80
4.5 PCI Devices on Bus 0 81

5 Appendix 82
5.1 Literature and WWW Resources 82
5.1.1 Bridges 82
5.1.2 PCI 82
5.1.3 PC•MIP 82
5.2 Board Revisions 83
5.3 Component Plans 84

Figures

Figure 1. Map of the Board - Top View 14
Figure 2. SO-DIMM DRAM Installation 19
Figure 3. Position of Flash on Bottom Side of B11 20
Figure 4. CompactFlash Card 21
Figure 5. Installation of a PC•MIP 23
Figure 6. Inserting the B11 RS232 SA Adapter 30
Figure 7. Installing the B11 RS232 SA Adapter Bezel 31
Figure 8. B11 with RS232 SA Adapter 31
Figure 9. Installing Standard SA Adapters 32
Figure 10. Position of Hex Switch 34
Figure 11. MENMON Address Mapping 44
Figure 12. MENMON - Power On Self Test Output with Self Test Message Mode EXTENDED 46
Figure 13. MENMON - Layout of the 0x41-Type Partition (PReP) 50
Figure 14. MENMON - Example PCI Configuration 63
Figure 15. B11 Interrupt Structure 78
Figure 16. Component Plan of B11 Rev. 02 - Top Side 84
Figure 17. Component Plan of B11 Rev. 02 - Bottom Side 84

Tables

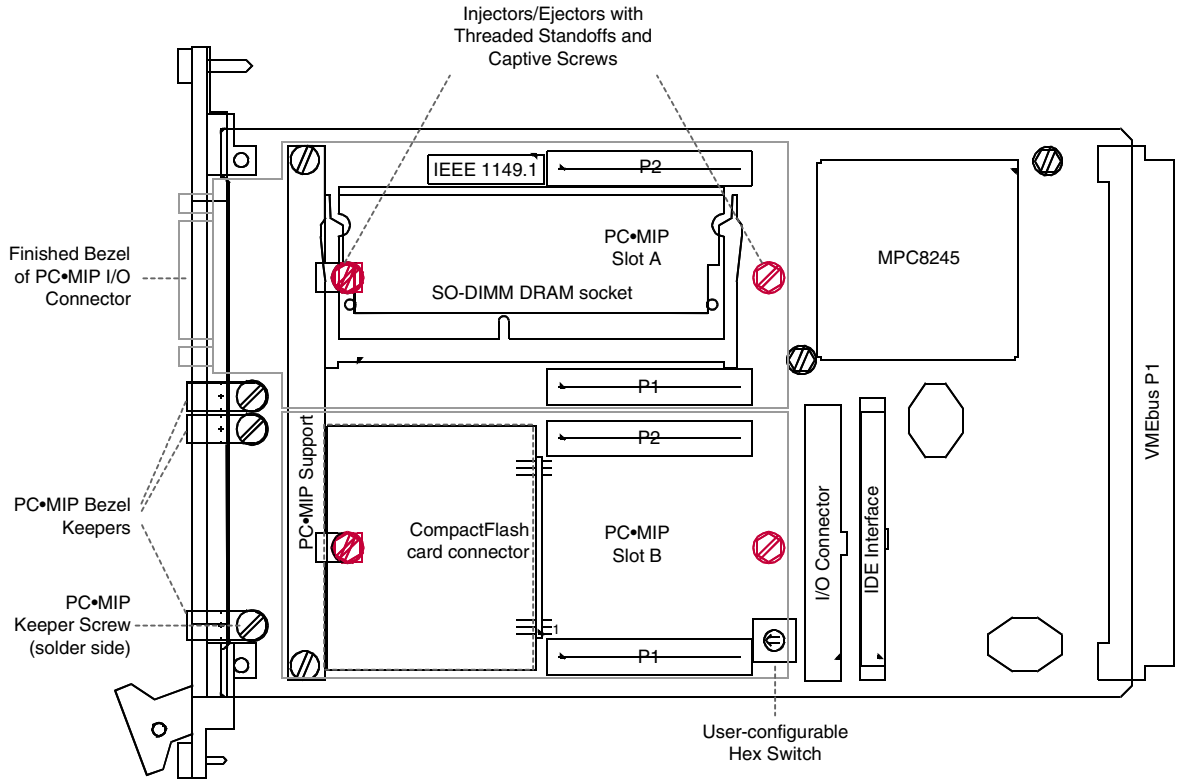
Table 1. Terminal Lines of the 9-pin D-Sub RS232 Plug Connector (COM1) . 16
Table 2. Pin Assignment of the 44-Pin IDE Connectors 25
Table 3. Signal Mnemonics for 40-pin and 44-pin IDE Connectors 26
Table 4. Pin Assignment of the 34-Pin I/O Connector. 27
Table 5. Signal Mnemonics of 34-pin I/O Connector 28
Table 6. Pin Assignment of the 9-pin D-Sub RS232 COM1 Plug Connector (08SA01-03) 29
Table 7. PCI Configuration Space Registers 37
Table 8. Pin Assignment of the 3-Row, 96-Pin VMEbus Connector P1 41
Table 9. Pin Assignment of the 16-pin IEEE 1149.1 Test Connector 42
Table 10. MENMON - Assignment for B11 Controller Devices. 49
Table 11. MENMON - Download Destination Devices. 57
Table 12. MENMON - Flash Sectors 58
Table 13. MENMON Command Overview 61
Table 14. Hex-Switch Settings 65
Table 15. MENMON System Calls - BRD_ID Fields 67
Table 16. MENMON System Calls - DSK_RD Fields 70
Table 17. MENMON VxWorks Bootline - List of Parameters and their Usage . 72
Table 18. MENMON - Common Parameters Passed by All MENMONs 73
Table 19. Memory Map - Processor View 74
Table 20. Address Mapping for PCI. 74
Table 21. BATS set up by MENMON 75
Table 22. PCI/ISA I/O Space Memory Map (addresses as seen from CPU) 76
Table 23. VMEbus Memory Mapping 77
Table 24. Interrupts on B11 79
Table 25. M1543 General Purpose Input (GPI) Pin Assignments 80
Table 26. M1543 General Purpose Input/Output (GPIO) Pin Assignments 80
Table 27. SMB 1 Devices. 80
Table 28. SMB 2 Devices. 80
Table 29. PCI Devices on Bus 0. 81
Table 30. Table of Hardware Revisions 83

1 Getting Started

This chapter will give an overview of the B11 and some hints for first installation in a VMEbus system as a "check list".

1.1 Map of the Board

Figure 1. *Map of the Board - Top View*



1.2 Configuring the Hardware

You should check your hardware requirements before installing the board in a system, since most modifications are difficult or even impossible to do when the board is mounted in an enclosure.

The following check list will give an overview on what you might want to configure.

☑ DRAM SO-DIMM module

The B11 is shipped without DRAM on the board. You should check on your main memory needs and install a suitable SO-DIMM module.

Refer to Chapter 2.5.1 SDRAM on page 19 for a detailed installation description and hints on supported SO-DIMM modules.

☑ CompactFlash

Refer to Chapter 2.5.3 CompactFlash on page 21 for a detailed installation description and hints on supported CompactFlash cards.

☑ Serial interface (SA) adapters

You can install standard serial interfaces such as RS232 (COM1) using MEN's SA adapters on the B11's 34-pin I/O connector.

Refer to Chapter 2.8.1 Serial Ports COM1/COM2 on page 29 for detailed installation descriptions.

☑ PC•MIPs

Refer to Chapter 2.6.1 Installing PC•MIPs on page 23 for a detailed installation description.

1.3 Integrating the Board into a System

You can use the following check list when installing the CPU board in a VMEbus system for the first time and with minimum configuration.

The board is completely trimmed on delivery. Perform the following procedure **without a PC•MIP installed**!

☑ Power-down the system.

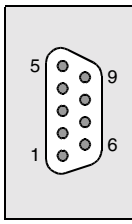
☑ Remove all boards from the VMEbus system.

☑ Plug an RS232 SA adapter into COM1 of the 34-pin I/O connector as described in Chapter 2.8.1 Serial Ports COM1/COM2 on page 29.

☑ Insert the B11 in slot 1 of your VMEbus system, making sure that the VMEbus connectors are properly aligned.

☑ Connect a terminal to the standard RS232 interface (COM1 via SA adapter with 9-pin D-Sub connector) by wiring the following lines to the connector:

Table 1. *Terminal Lines of the 9-pin D-Sub RS232 Plug Connector (COM1)*



5	GND	9	-
4	-	8	-
3	TXD	7	-
2	RXD	6	-
1	-		

☑ Set your terminal to the following protocol:

- 9600 baud data transmission rate
- 8 data bits
- 1 stop bit
- no parity

☑ Power-up the system.

☑ The terminal displays the following message:

```
_____________________ Secondary MenMon for the B011 Version 1.0 _______________
|                                                                               |
|          (c) 1999 - 2000 MEN mikro elektronik GmbH Nuernberg                  |
|             Parts of this code based on Motorola's Dink32                     |
|                  Created  Apr 25 2001     16:38:09                           |
|_______________________________________________________________________________|
|    HW Revision: 00.00.00            |            CPU: MPC8245     (MAP B)    |
|  Serial Number: 2                   |  CPU/MEM Clock: 300 / 100 MHz          |
|    Board Model: B011-00             |    DIMM Module:  64 MB Setup: 222      |
 \_____________________________________________________________________________/
Init VME controller regs @0x81800000 (VME slot1 function enabled)

press 'ESC' to setup/MENMON
Selftest running ...
CHECKSUM              ==> OK

*** Can't jump to bootstrapper. BS address in EEPROM invalid!
MenMon>
```

☑ Now you can use the MENMON debugger (see detailed description in Chapter 3 MENMON on page 43).

☑ Observe the installation instructions for the respective software.

1.4 Installing Operating System Software

The B11 supports VxWorks, Linux and OS-9.

By standard, no operating system is installed on the board. Please refer to the respective documentation on how to install software and set up the bootstrapper!

2 Functional Description

The following describes the individual functions of the B11 and their configuration on the board. There is no detailed description of the individual controller chips and the CPUs. They can be obtained from the data sheets or data books of the semiconductor manufacturer concerned (Chapter 5.1 Literature and WWW Resources on page 82).

2.1 Power Supply

The B11 is supplied with +5V via the VMEbus. For the PC•MIP modules ±12V are also provided via the VMEbus backplane.

The onboard power supply generates the core voltage supply for the PowerPC, which is 2.5V.

2.2 Clock Supply

The clock supply generates all clocks for the on-board devices (PowerPC, SDRAM, host bridge, PCI bus devices). The clock frequency is factory-set.

The local PCI clock is limited to 33MHz.

2.3 PowerPC CPU

The B11 is equipped with the MPC8245 Kahlua processor, which includes a 32-bit superscalar PowerPC 603e core and the MPC107 host-to-PCI bridge.

2.3.1 General

The PowerPC architecture, developed jointly by Motorola, IBM, and Apple Computer, is based on the POWER architecture implemented by the RS/6000™ family of computers. The PowerPC architecture takes advantage of recent technological advances in such areas as process technology, compiler design, and RISC microprocessor design to provide software compatibility across a diverse family of implementations, primarily single-chip microprocessors, intended for a wide range of systems.

2.3.2 Heat Sink

A heat sink is provided to meet thermal requirements.

Note: MEN gives no warranty on functionality and reliability of the B11 if you use any other processor or heat sink than that supplied by MEN. Please contact either MEN directly or your local MEN sales office!

2.4 Bus Structure

2.4.1 Host-to-PCI Bridge

The MPC107 (internal in MPC8245) is used as host bridge and memory controller for the PowerPC processor. All transactions of the PowerPC to the PCI bus are controlled by the host bridge. The SDRAM and boot Flash are connected to the local memory bus of the MPC107.

The B11 supports concurrent transfers on PowerPC and PCI buses.

The PCI interface is PCI bus Rev. 2.1 compliant and supports all bus commands and transactions. Master and target operations are possible. Only big-endian operation is supported.

2.4.2 Local PCI Bus

The local PCI bus is controlled by the MPC107 host-to-PCI bridge. It runs at 33MHz. The I/O voltage is fixed to 3.3V. The data width is 32 bits.

Major functional elements of the B11, such as PC•MIP extension, are connected on the local PCI bus.

2.4.3 PCI-to-ISA Bridge

The M1543 is the "Southbridge" between the PCI and ISA bus, providing full PCI and ISA compatible functions. The M1543 provides integrated Super I/O (2 serial ports/1 parallel port), system peripherals (ISP) (2 82C59 and serial interrupt, 1 82C54), advanced features (type F and distributed DMA) in the DMA controller (2 82C37), PS2 keyboard/mouse controller, 2-channel dedicated IDE master controller with Ultra-33 specification and System Management Bus (SMB).

M1543 also provides a PCI-to-ISA IRQ routing table, and level-to-edge trigger transfer. The chip provides two additional IRQ lines and one programmable chip-select. The interrupt lines can be routed to any of the available ISA interrupts.

2.4.4 PCI-to-VMEbus Bridge

The B11 has a PCI-to-VME bridge for connection to the VMEbus. It is controlled by an FPGA.

2.5 Memory

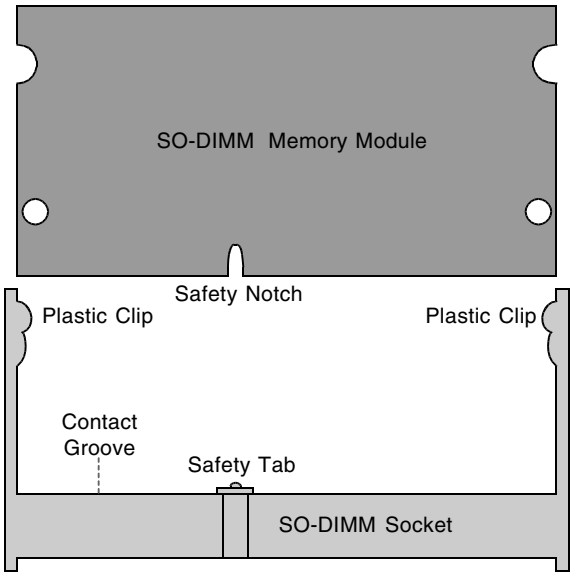
2.5.1 SDRAM

One SDRAM bank (bank 0) is implemented on B11. Bank 0 is connected to a 144-pin SO-DIMM connector for easy extension. The MPC107 can handle SDRAM devices with up to 8x128Mbit DRAM.

2.5.1.1 Installing SO-DIMM DRAM

The B11 is shipped without any DRAM SO-DIMM module installed. To install a SO-DIMM module, please stick to the following procedure.

Figure 2. *SO-DIMM DRAM Installation*



The DRAM module will only fit as shown above because of a safety tab on the SO-DIMM socket which requires a notch in the SO-DIMM module.

- ☑ Power down the system before installing a SO-DIMM module to avoid damage of the B11!
- ☑ Remove the B11 from the system.
- ☑ Remove the PC•MIP support from the B11. To do this, loosen and remove the two support screws at the bottom side of the B11. (See Figure 5, Installation of a PC•MIP, on page 23.)
- ☑ Place the memory module into the socket at a 45° angle and make sure that the safety tab and notch are aligned.
- ☑ Carefully push the memory module into the contact groove of the socket.
- ☑ Press the memory module down until it clicks into place.
- ☑ The plastic clips of the socket now hold the memory module in place.

☑ Reinstall the PC•MIP support. To do this, fasten the support using the two support screws at the bottom side of the B11.

☑ To remove the DRAM module you must again remove and then reinstall the PC•MIP support.

☑ To release the module, squeeze both plastic clips outwards and carefully pull the module out of the socket.

2.5.1.2 Supported SO-DIMM Modules

You can install standard SO-DIMM modules with SDRAM components. See MEN's website for memory modules available from MEN.

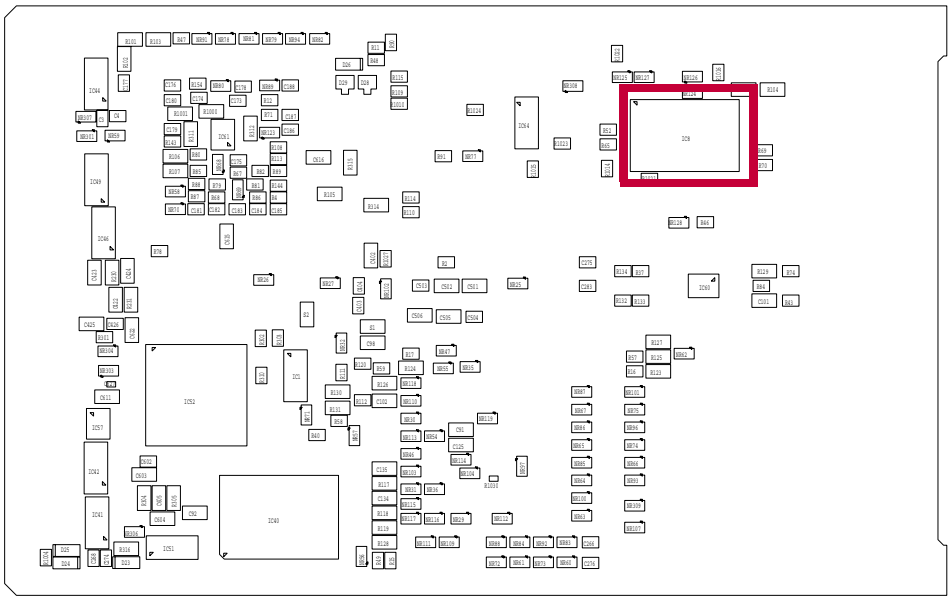
Note: MEN gives no warranty on functionality and reliability of the B11 if you use any other module than that qualified and/or supplied by MEN. Please contact either MEN directly or your local MEN sales office.

2.5.2 Flash

The B11 has on-board Flash. It is controlled by the MPC107 host bridge and can accommodate 2MB. The data bus is 8 bits wide.

Flash memory contains the boot software for the MENMON/operating system bootstrapper and application software. The MENMON sectors are software-protected against illegal write transactions through a password in the serial download function of MENMON (cf. Chapter 3.6 Updating Flash Devices on page 57).

Figure 3. *Position of Flash on Bottom Side of B11*



2.5.3 CompactFlash

CompactFlash is a standard for small form factor ATA Flash drives. It is electrically compatible to the PC Card 1995 and PC Card ATA standards.

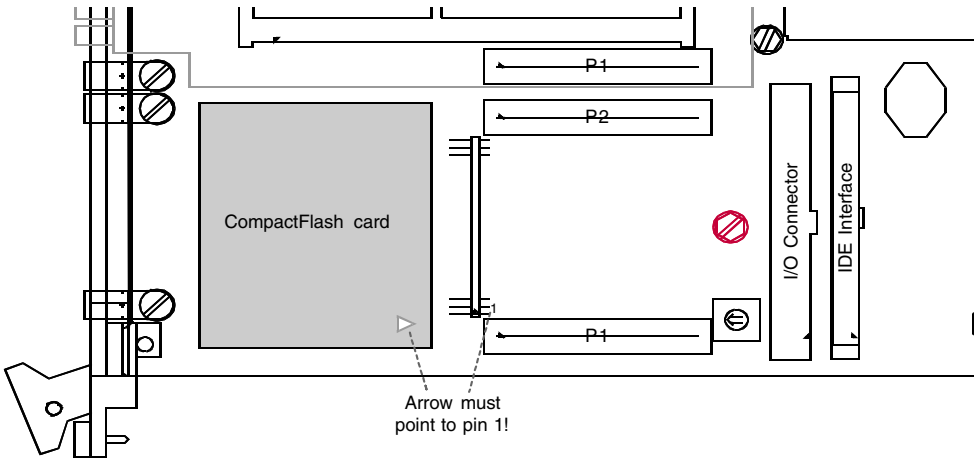
The CompactFlash standard is supported by industry's leading vendors of Flash cards.

You can use CompactFlash cards with the B11 through a connector residing within PC•MIP slot B, i. e. the connector is located under the PC•MIP module, if you have installed one. CompactFlash cards are operated in a True IDE Mode.

2.5.3.1 Installing CompactFlash

The B11 is shipped without a CompactFlash card installed. To install CompactFlash, please stick to the following procedure.

Figure 4. *CompactFlash Card*



☑ Power down your system and remove the B11 from the system.

☑ If there is a PC•MIP module installed in slot B, remove the PC•MIP module.

☑ Remove the PC•MIP support from the B11. To do this, loosen and remove the two support screws at the bottom side of the B11. (See Figure 5, Installation of a PC•MIP, on page 23.)

☑ Insert the card carefully as indicated by the arrow on top of the card, making sure that all the contacts are aligned properly and the card is firmly connected with the card connector.

Note: Make sure to align the arrow on the CompactFlash card with pin 1 of the card connector, i.e. the **arrow on the CompactFlash card faces the PCB**.

☑ Reinstall the PC•MIP support. To do this, fasten the support using the two support screws at the bottom side of the B11.

☑ To remove the CompactFlash card you must again remove and then reinstall the PC•MIP and PC•MIP support as described above.

☑ Observe manufacturer notes on usage of CompactFlash cards.

2.5.3.2 Supported CompactFlash Cards

The B11 supports standard CompactFlash cards. See MEN's website for CompactFlash cards available from MEN.

2.5.4 EEPROM

The B11 has a 2-Kbyte serial EEPROM for factory data, MENMON parameters, and for the VxWorks bootline.

2.6 PC•MIP Slots

The B11 has two PC•MIP slots for Type-I and Type-II modules. The PC•MIPs are connected to the local PCI bus.

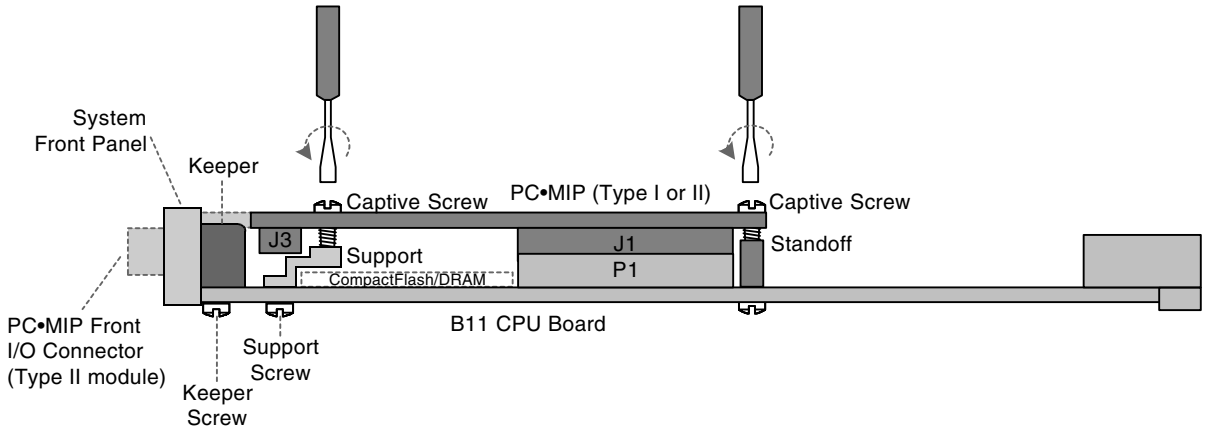
The PC•MIP slots enable the user to add functionality to the B11 CPU board, from graphics to process I/O.

2.6.1 Installing PC•MIPs

Perform the following steps to install a PC•MIP:

- ☑ Power down your system and remove the B11 from the system.
- ☑ If you want to install a Type-II PC•MIP (with front connector), you must remove the blank bezel at the front panel of the B11 first:
  Remove the respective bezel keeper by loosening the keeper screw at the bottom side of the B11. (See Figure 1, Map of the Board - Top View, on page 14).
- ☑ Place the finished bezel supplied with your PC•MIP in the front panel cut-out and reinstall the bezel keeper.
- ☑ If you are installing a Type-II PC•MIP, carefully put the module's front connector through the finished bezel, holding the module at a 45° angle.
- ☑ Place the PC•MIP on the target slot of the B11, aligning the two connectors (P1/J1, P2/J2) and the two standoffs.
- ☑ Screw the PC•MIP to the carrier by **alternately** tightening the two captive screws on the label side of the PC•MIP. The module will be "injected" safely.

Figure 5. *Installation of a PC•MIP*



To deinstall PC•MIPs from the carrier board, just loosen the appropriate screws at the label side of the PC•MIP. The injector/ejector system will "eject" the PC•MIP.

If you want to install or deinstall a CompactFlash card or DRAM module, you must remove the PC•MIP support from the B11. To do this, loosen and remove the two support screws at the bottom side of the B11.

2.6.2 PC•MIP Connectors

PC•MIP modules connect to the B11's PCI bus via the two identical 64-pin connectors P1 and P2. The connector layout is fully compatible to the PC•MIP specification and will not be repeated here.

Please note that the B11 has no third 64-pin connectors (P3), and therefore does not support rear I/O connection.

Connector types of P1 and P2:

- 64-pin SMT plug connector according to IEEE P1386, e. g. Molex 71436-0864
- Mating connector:
  64-pin SMT receptacle connector according to IEEE P1386, e. g. Molex 71439-1864

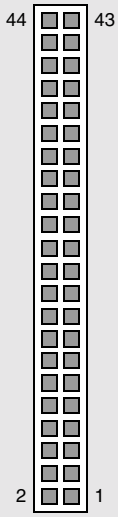
2.7 IDE Interface

The B11 provides a 44-pin plug for IDE connection. The pinning of this connector is compliant with the ATA-4/ATAPI specification.

Connector Types 44-pin connectors:

- 44-pin, 2-row SMT plug, 2mm pitch
- Mating connector:
  44-pin, 2-row IDC receptacle, 2mm pitch

Table 2. *Pin Assignment of the 44-Pin IDE Connectors*



	44	GND	43	GND
	42	+5V	41	+5V
	40	GND	39	IDE_RACT#
	38	IDE_RCS3#	37	IDE_RCS1#
	36	IDE_RA[2]	35	IDE_RA[0]
	34	-	33	IDE_RA[1]
	32	-	31	IDE_RIRQ
	30	GND	29	IDE_RDAK#
	28	GND	27	IDE_RRDY#
	26	GND	25	IDE_RRD#
	24	GND	23	IDE_RWR#
	22	GND	21	IDE_RDRQ
	20	-	19	GND
	18	IDE_RD[15]	17	IDE_RD[0]
	16	IDE_RD[14]	15	IDE_RD[1]
	14	IDE_RD[13]	13	IDE_RD[2]
	12	IDE_RD[12]	11	IDE_RD[3]
	10	IDE_RD[11]	9	IDE_RD[4]
	8	IDE_RD[10]	7	IDE_RD[5]
	6	IDE_RD[9]	5	IDE_RD[6]
	4	IDE_RD[8]	3	IDE_RD[7]
	2	GND	1	IDE_RRST#

Table 3. *Signal Mnemonics for 40-pin and 44-pin IDE Connectors*

Signal	Direction	Function
+5V	-	+5V power supply, current-limited to 1.5A by a fuse
GND	-	Digital ground
IDE_RA[2:0]	out	IDE address [2:0]
IDE_RACT#	in	IDE active
IDE_RCS1#	out	IDE chip select 1
IDE_RCS3#	out	IDE chip select 3
IDE_RD[15:0]	in/out	IDE data [15:0]
IDE_RDAK#	out	IDE DMA acknowledge
IDE_RDRQ	in	IDE DMA request
IDE_RIRQ	in	IDE interrupt request
IDE_RRD#	out	IDE read strobe
IDE_RRDY#	in	IDE ready
IDE_RRST#	out	IDE Reset
IDE_RWR#	out	IDE write strobe

2.8 I/O Connector

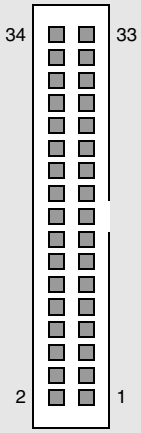
The B11 features a 34-pin I/O connector that implements several interfaces:

- Two serial ports COM1/COM2 (compatible with MEN's SA adapters, see Chapter 2.8.1 Serial Ports COM1/COM2 on page 29)
- A reset and abort button[1]
- Two user-configurable LEDs[1]
- Keyboard/mouse[2]
  The built-in PS2/AT keyboard and PS2 mouse controller of the M1543 is connected to the I/O connector.
- USB port[2]

Connector Types 34-pin connector:

- 34-pin low-profile plug, 2.54mm pitch, for ribbon-cable connection
- Mating connector:
  34-pin IDC receptacle, e.g. Elco Series 8290 IDC socket

Table 4. *Pin Assignment of the 34-Pin I/O Connector*



34	USB+	33	USB-
32	+5V	31	GND
30	MSDATA	29	MSCLK
28	KBDATA	27	KBCLK
26	LED2	25	LED1
24	ABRTBTN#	23	PWRBTN#
22	+5V	21	GND
20	RI2#	19	DCD2#
18	CTS2#	17	DSR2#
16	RTS2#	15	DTR2#
14	RXD2	13	TXD2
12	+5V	11	GND
10	RI1#	9	DCD1#
8	CTS1#	7	DSR1#
6	RTS1#	5	DTR1#
4	RXD1	3	TXD1
2	+5V	1	GND

[1] These ports are implemented on the connector, but there are no buttons and LEDs on the B11 board. Please turn to our sales staff if you need any help or extensions to use these interfaces.

[2] These ports are implemented on the connector but may not be supported through software. Please turn to our sales staff if you need any help or extensions to use these interfaces.

Table 5. *Signal Mnemonics of 34-pin I/O Connector*

	Signal	Direction	Function
Power	+5V	-	+5V power supply
	GND	-	Digital ground of respective interface
Mouse/ Keyboard	KBDATA	out	Keyboard data
	KBCLK	out	Keyboard clock
	MSDATA	out	Mouse data
	MSCLK	out	Mouse clock
LEDs	LED1	out	LED1 cathode[1]
	LED2	out	LED2 cathode[1]
Button	ABRTBTN#	in	Abort button[2]
	PWRBTN#	in	Reset button[2]
COM2	CTS2#	in	Serial port 2 clear to send
	DCD2#	in	Serial port 2 data carrier detect
	DSR2#	in	Serial port 2 data set ready
	DTR2#	out	Serial port 2 data terminal ready
	RI2#	in	Serial port 2 ring indicator
	RTS2#	out	Serial port 2 request to send
	RXD2	in	Serial port 2 receive data
	TXD2	out	Serial port 2 transmit data
COM1	CTS1#	in	Serial port 1 clear to send
	DCD1#	in	Serial port 1 data carrier detect
	DSR1#	in	Serial port 1 data set ready
	DTR1#	out	Serial port 1 data terminal ready
	RI1#	in	Serial port 1 ring indicator
	RTS1#	out	Serial port 1 request to send
	RXD1	in	Serial port 1 receive data
	TXD1	out	Serial port 1 transmit data
USB	USB+, USB-	in/out	USB port differential pair

[1] Connect the anode to +5V (pin 22 of 34-pin connector).
[2] Connect the button's second terminal to GND (pin 21 of 34-pin connector).

2.8.1 Serial Ports COM1/COM2

The B11 provides two high-performance 16550 compatible UARTs with 16-byte send/receive FIFOs and programmable baud rate generator. You can set the baud rate through MENMON.

The signal level is fixed to TTL. This allows flexible line interface configuration using serial interface (SA) adapters. MEN offers a mounting kit for connection of their standard SA adapters (see MEN's website).

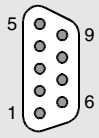
There is also a specialized SA adapter available for RS232 connection (see MEN's website).

Note: MEN gives no warranty on functionality and reliability of the B11 and SA adapters used if you install SA adapters in a different way than described in this manual.

2.8.1.1 Installing the B11 RS232 SA Adapter

With the help of MEN's SA01-03 adapter you can easily lead COM1 to the front panel, e.g. to connect a terminal, using a standard 9-pin D-Sub connector. The connector fits in the PC•MIP cut-outs at the front panel of the B11, so that you do not need a second front panel. By consequence, if you connect the adapter through a PC•MIP cut-out, you cannot use the respective PC•MIP slot.

Table 6. *Pin Assignment of the 9-pin D-Sub RS232 COM1 Plug Connector (08SA01-03)*



5	GND	9	RI-O
4	DTR-O	8	CTS-O
3	TXD-O	7	RTS-O
2	RXD-O	6	DSR-O
1	DCD-O		

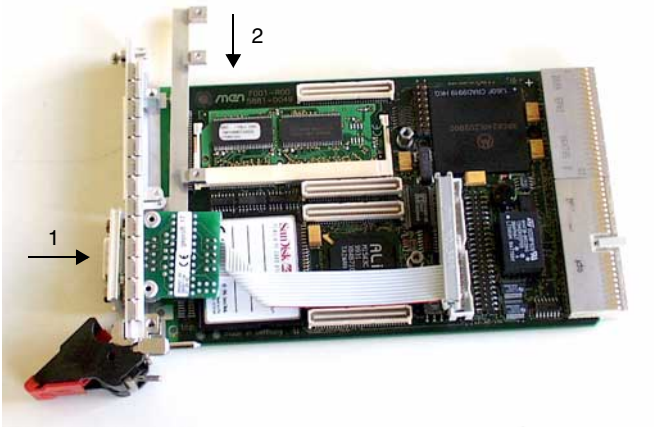
Connector types:

- 9-pin D-Sub plug according to DIN41652/MIL-C-24308, with thread bolt UNC 4-40
- Mating connector:
  9-pin D-Sub receptacle according to DIN41652/MIL-C-24308, available for ribbon cable (insulation piercing connection), hand-soldering connection or crimp connection

Perform the following steps to install the SA01-03:

- ☑ Power down your system and remove the B11 from the system.
- ☑ If there is a PC•MIP module installed in slot B, remove the PC•MIP module.
- ☑ Remove the PC•MIP support from the B11. To do this, loosen and remove the two support screws at the bottom side of the B11. (See Figure 5, Installation of a PC•MIP, on page 23.)
- ☑ Remove the bezel keeper of PC•MIP slot B at the front panel of the B11 by loosening the keeper screw at the bottom side of the B11. (See Figure 1, Map of the Board - Top View, on page 14).
- ☑ Remove the blank bezel of PC•MIP slot B.
- ☑ Put the ribbon cable connector through the front panel cut-out, then put the whole SA adapter through the front panel cut-out. (See following figure, step "1".)
- ☑ Carefully put the PC•MIP support back in its place as shown in the following figure, step "2".

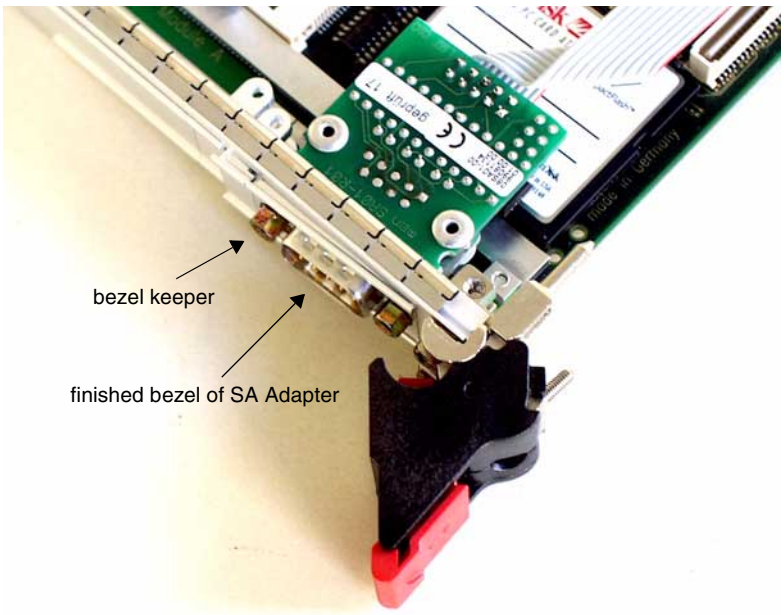
Figure 6. *Inserting the B11 RS232 SA Adapter*[1]



- ☑ Screw the PC•MIP support tightly to the B11 using the two support screws at the bottom side of the B11.
- ☑ Align the finished bezel of the SA adapter with the bezel keeper removed before, and reinstall the bezel keeper.

[1] The figure shows MEN's F1 board, which is mechanically identical for PC•MIP and SA adapters.

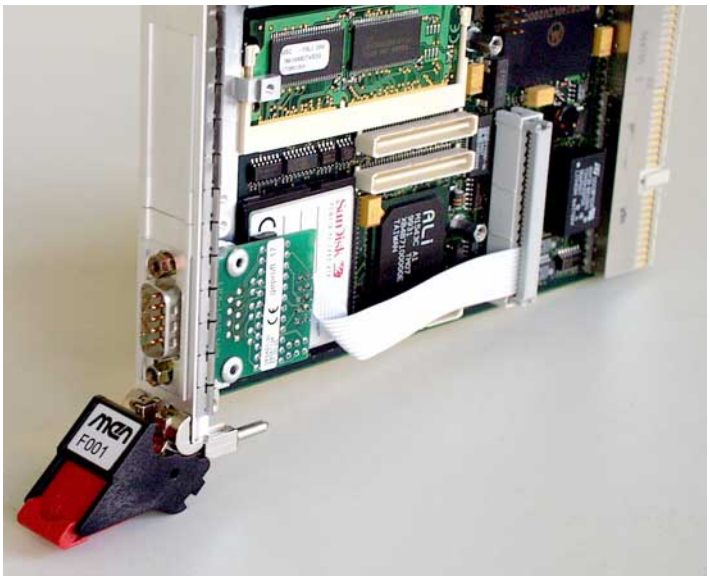
Figure 7. *Installing the B11 RS232 SA Adapter Bezel*



☑ Insert the ribbon cable connector into the B11's 34-pin I/O connector, making sure that all the contacts are aligned properly and the connectors are firmly linked.

☑ To remove the SA adapter, follow the above procedure in reverse order.

Figure 8. *B11 with RS232 SA Adapter*



2.8.1.2 Installing Standard SA Adapters

Both COM1 and COM2 support the use of any of MEN's standard SA adapters. This allows you to choose from a number of available line interfaces, from RS232 to RS422/RS485 to TTY, with or without optical isolation.

For mechanical reasons, the standard SA adapters do not fit into the smaller PC•MIP cut-outs. Therefore MEN offers an additional 1-slot front panel with two SA adapter cut-outs, including a ribbon cable for connection of both COM1 and COM2.

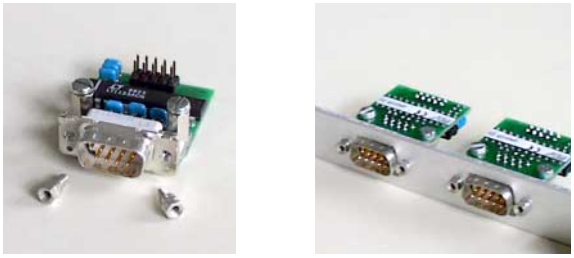
Note: If you use a second front panel for SA adapters, you can use both PC•MIP slots.

See MEN's website for compatible adapters and ordering numbers.

Perform the following steps to install standard SA adapters using MEN's mounting kit:

- ☑ Power-down your system and remove the B11 from the system.
- ☑ Remove the front panel screws of the SA adapter.
- ☑ Use the front panel screw to fasten the SA adapter at the additional SA adapter front panel.

Figure 9. *Installing Standard SA Adapters*



- ☑ Plug the prefolded ribbon cable to the 34-pin I/O connector on the B11.
- ☑ Plug the two 10-pin connectors of the ribbon cable to the SA adapter connectors.
- ☑ Make sure to always match the pins correctly (pin 1 is marked by a triangle on the ribbon cable connector).
- ☑ You can now reinsert the B11 and the additional front panel into your system. Make sure to fasten the SA adapter front panel appropriately in your enclosure!

2.9 Hardware Monitor

The LM75 hardware monitor is used for temperature management. It continuously measures the on-board temperature.

2.10 Real-Time Clock and NVRAM

The B11 includes the 41T56 SMB real-time clock with integrated NVRAM. A local GoldCap capacitor supplies the backup voltage. The real-time clock must be supplied via the VMEbus *STANDBY* line.

The 56-byte NVRAM is organized as a 56 bytes x 8 bits SRAM.

2.11 Watchdog

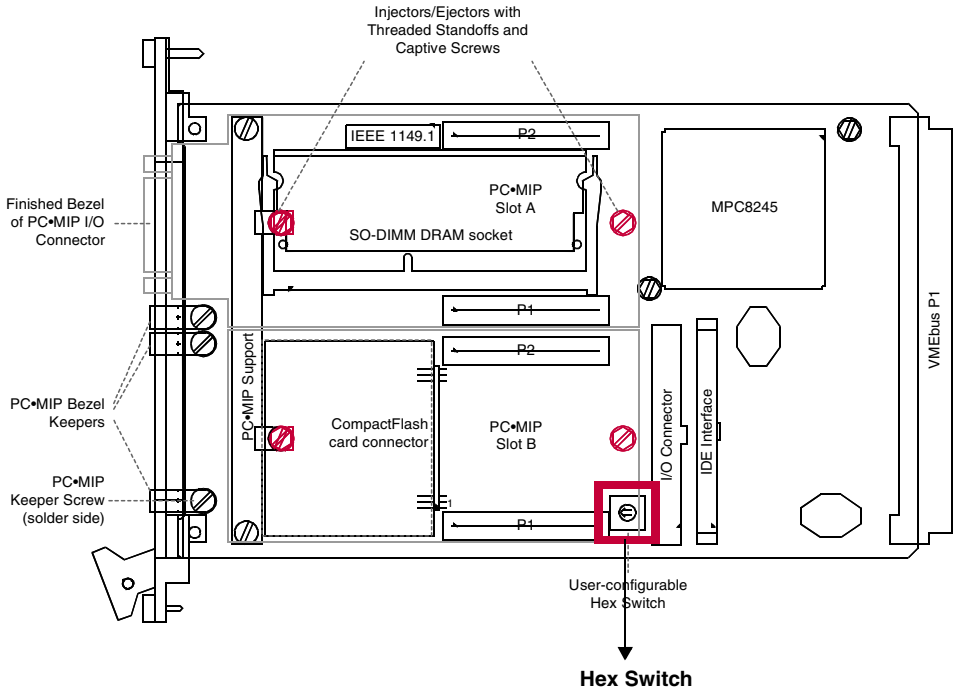
The B11 uses an SMS24 watchdog, which has three functions:

- Power-On Reset
- Watchdog
- EEPROM (2Kbytes) (see Chapter 2.5.4 EEPROM on page 22)

2.12 User-Defined Hex Switch

The B11 provides a rotary hex switch for operating system requirements and user applications. Please refer to the corresponding software manual for the implemention.

Figure 10. *Position of Hex Switch*



Note: The hex switch is not available on B11 hardware revisions 00.xx.xx.

2.13 VMEbus Interface

The B11's VMEbus interface conforms to the VMEbus specification. It has the following features:

- Slot-1 functionality
- Wide range of VMEbus address and data transfer modes
  - Master D08(EO):D16:A24:A16:RMW[1]
  - Slave D08(EO):D16:A24:BLT
- Interrupt handler: 7-level, D08(O):IH(7-1)
- Interrupter: 7-level, D08(O):I(7-1):ROAK

2.13.1 SRAM

The SRAM is accessible from the VMEbus, the VME address can be configured in steps of up to 1MB in the SLV - Slave Control Register (0x14) (r/w). Access from the VMEbus can be disabled.

The SRAM has a size of 1MB. It is accessible via block and standard transfers (user and supervisor space).

See also Table 23, VMEbus Memory Mapping, on page 77.

2.13.2 Slot-1 Function

The slot-1 function is auto-detected. It can be read from the SYSCTL register (bit *SYSCON*).

2.13.3 Interrupter

The B11 has one interrupter. It can generate an interrupt on all seven levels, which can be configured through bits *ILx* in the INTR - VME Interrupter Control Register (0x00) (r/w). The interrupt vector is written to register INTID - VME Interrupt STATUS/ID Register (0x04) (r/w). An interrupt request is generated by setting bit *INTEN* in the INTR register. This bit is cleared in the IACK cycle.

2.13.4 Interrupt Handler

The B11 can receive interrupts on all seven levels. In addition, it can handle *ACFAIL#* interrupts. You can mask interrupts through the IMASK - Interrupt Mask Register (0x0C) (r/w).

If a VME interrupt occurs that is not masked, the PCI-to-VME bridge generates a PCI interrupt (routed to *INT_B* on B11). Then, the software must read the ISTAT - Interrupt Status Register (0x08) (r/w) to detect which VME interrupts are pending. The ISTAT register will only show bits that were enabled in IMASK!

The interrupt vector must then be fetched through a read to the VME IACK space. The address within the IACK space must reflect the VMEbus level (word access 0xA or byte access 0xB for level 5).

Naturally, there is no vector for *ACFAIL#* interrupts. To reset such interrupts, write 1 to the *ACFST* bit in ISTAT - Interrupt Status Register (0x08) (r/w).

[1] For RMW cycles PCI to VME only byte/word (8-bit/16-bit) accesses are allowed!

2.13.5 Bus Errors

If a bus error occurs, bit *BERR* in the MSTR - Master Control Register (0x10) (r/w) is set. An interrupt is triggered if the *IBERREN* Bit in the MSTR - Master Control Register (0x10) (r/w) is set. The timeout for *BERR* is 60µs. The bus error can be cleared by writing 1 to the *BERR* bit.

2.13.6 Master Access

Master access is always via

- 0x39 = standard user data access for A24:D16, standard I/O
- 0x29 = short user access for A16:D16, short I/O

2.13.7 Atomic Operations

CPU-to-SRAM Operations

Not supported.

CPU-to-VME Operations

Read-Modify-Write operations to the VMEbus can be done via bit *RMW* in the MSTR - Master Control Register (0x10) (r/w). Only byte and word accesses are allowed, with word accesses being made to even addresses.

VME-to-SRAM Operations

Not supported.

2.13.8 PCI Configuration Space Registers

The Configuration Registers from 0x00 to 0x3C conform with the PCI Device Configuration Header Format.

Table 7. *PCI Configuration Space Registers*

Address	Byte			
	3	2	1	0
0x00	Device ID (0x5056)		Vendor ID (0x1172)	
0x04	Status Register		Command Register	
0x08	Class Code (0x068000)			Revision ID (currently 0x01)
0x0C	BIST	Header Type	Latency Timer	Cache Line Size
0x10	Base Address Register 0			
0x14	*This register is always 0.*			
0x18	*This register is always 0.*			
0x1C	*This register is always 0.*			
0x20	*This register is always 0.*			
0x24	*This register is always 0.*			
0x28	Card Bus CIS Pointer			
0x2C	Subsystem ID		Subsystem Vendor ID	
0x30	Expansion ROM Base Address Register			
0x34	Reserved			
0x38	Reserved			
0x3C	Maximum Latency	Minimum Grant	Interrupt Pin	Interrupt Line

2.13.9 Runtime Registers

The registers are not directly accessible from the VMEbus but from the PCI bus. Writes to registers can only be done through the PCI bus (read/write registers) or from the bridges internal modules. Read only registers can not be written from the PCI bus, only from the internal modules.

The address of these registers can be computed by reading out BAR0 of the PCI-to-VME bridge's PCI configuration space. Then add 0x01800000 plus the register offset to this value.

INTR - VME Interrupter Control Register (0x00) (r/w)

7..4	3	2	1	0
-	INTEN	IL2	IL1	IL0

This register controls the internal interrupter. Interrupt levels from 1 to 7 can be set. The interrupt is generated only when the INTEN bit is set.

INTEN 0 = interrupt disabled (default)
1 = enable interrupt at level specified through *ILx*

ILx The interrupt level is set in binary code (e. g. *ILx*=011 is IRQ3).
default: 0x0

INTEN should be set after the *ILx* bits are set to avoid glitches on the IRQ lines.

INTEN is automatically cleared during the acknowledge cycle and the request is removed (ROAK). The *ILx* bits, however, remain set until they are overwritten.

Check the *INTEN* bit to verify that the interrupt has been acknowledged.

INTID - VME Interrupt STATUS/ID Register (0x04) (r/w)

7..0
INT_ID

In this Register, the STATUS/ID of the internal interrupter is set.

INT_ID The STATUS/ID of the interrupt that the external handler reads during the IACK cycle.
default: 0x00

ISTAT - Interrupt Status Register (0x08) (r/w)

7	6	5	4	3	2	1	0
I7	I6	I5	I4	I3	I2	I1	ACFST

Ix The PCI Master can read the asserted interrupts here if they are not masked in the *IMASK* register. Interrupts are reset automatically if the external interrupter removes its request. Writes to these bits are ignored.

ACFST If this reads 1, an *ACFAIL#* interrupt has occurred. This interrupt is stored, because it is not static. It can be cleared by writing 1.
default: 0

In order to read the interrupt STATUS/ID from the interrupter, the CPU must generate a read cycle to the IACK memory area by setting the last significant address bits *A[3..1]* to the corresponding interrupt level (e. g. for *IRQ5#* set *A[3:1]* to [101]).

IMASK - Interrupt Mask Register (0x0C) (r/w)

7	6	5	4	3	2	1	0
IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	ACFEN

IENx If the corresponding interrupt pin IRQx is asserted, the bridge will signal an interrupt to the PCI side (*INTA#*).
0 = Mask interrupt (default)
1 = Enable interrupt

ACFEN When this bit is set, and an *ACFAIL#* is detected, an interrupt on the PCI side is generated.
0 = Mask interrupt (default)
1 = Enable *ACFAIL#* interrupt

MSTR - Master Control Register (0x10) (r/w)

7..5	4	3	2	1	0
-	POSTWR	IBERREN	BERR	REQ	RMW

POSTWR Posted Write Access to VMEbus
0 = Delayed write access to VMEbus (default)
1 = Posted write access to VMEbus

IBERREN Interrupt Bus Error Enable
This bit enables an interrupt to PCI if a VMEbus *BERR#* signal occurs. This interrupt is level-triggered!
0 = Disable interrupt (default)
1 = Enable interrupt

BERR Monitor for VMEbus *BERR#* signal.
0 = No VMEbus error (default)
1 = VMEbus error occurred, cleared by writing 1

REQ Set VMEbus requester scheme
0 = Release On Request (ROR) (default)
1 = Release When Done (RWD)

If this bit is changed from 0 to 1, i. e. from ROR to RWD, and there were previous accesses over the master interface, it is recommended to do a dummy read to free the bus.

RMW Enable single Read-Modifiy-Write-Cycle
0 = Normal cycle (default)
1 = RMW cycle. Master keeps *AS#* asserted during back-to-back read/ write cycle.

This bit is automatically cleared after the RMW cycle and must be set for the next RMW cycle again.

Note: During RMW cycles all interrupts on the host CPU should be masked.

SLV - Slave Control Register (Øx14) (r/w)

7..5	4	3..0
-	SLEN	SLBASE

SLEN 0 = Slave Unit disabled (default)
1 = Slave Unit enabled

SLBASE The slave's base address. Specifies the lowest address in the VMEbus address range that will be decoded. This field will be compared with the VME address *A[23:20]*. Since only *A[23:20]* are monitored, the smallest possible address space is 1MB.
default: 0000

SYSCTL - System Controller Register (Øx18) (r/w)

7..3	2	1	0
-	ATO	SYSRES	SYSCON

ATO Monitor for Arbitration Timeout Signal
0 = No Arbitration Timeout (default)
1 = Arbitration Timeout occurred , cleared by writing 1

SYSRES Reset VMEbus
0 = No assertion of *SYSRES#* (default)
1 = Assert output pin *SYSRES#*

SYSCON If set to 1, the system controller unit is enabled. This is only allowed if the board resides in VMEbus slot 1. The state after reset depends on successful automatic detection of the board's location. If this bit is set after reset but the board is not in slot 1, detection fails and the bit must be cleared manually.

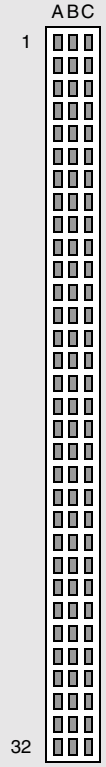
When set, the Arbiter, Arbitration Timer, Bus Timer and IACK-Daisy-Chain Driver are enabled.

2.13.10 Connection

Connector types:

- Type-C 96-pin plug connector according to DIN41612/MIL-C-55302/IEC603-2
- Mating connector:
  Type-C 96-pin receptacle according to DIN41612/MIL-C-55302/IEC603-2

Table 8. *Pin Assignment of the 3-Row, 96-Pin VMEbus Connector P1*



	A	B	C
1	D0	BBSY#	D8
2	D1	-	D9
3	D2	ACFAIL#	D10
4	D3	BG0IN#	D11
5	D4	BG0OUT#	D12
6	D5	BG1IN#	D13
7	D6	BG1OUT#	D14
8	D7	BG2IN#	D15
9	GND	BG2OUT#	GND
10	SYSCLK	BG3IN#	-
11	GND	BG3OUT#	BERR#
12	DS1#	-	SYSRESET#
13	DS0#	-	LWORD#
14	WRITE#	-	AM5
15	GND	BR3#	A23
16	DTACK#	AM0	A22
17	GND	AM1	A21
18	AS#	AM2	A20
19	GND	AM3	A19
20	IACK#	GND	A18
21	IACKIN#	-	A17
22	IACKOUT#	-	A16
23	AM4	GND	A15
24	A7	IRQ7#	A14
25	A6	IRQ6#	A13
26	A5	IRQ5#	A12
27	A4	IRQ4#	A11
28	A3	IRQ3#	A10
29	A2	IRQ2#	A9
30	A1	IRQ1#	A8
31	-12V	-	+12V
32	+5V	+5V	+5V

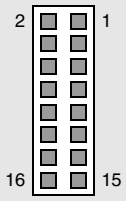
2.14 IEEE 1149.1 (JTAG)/COP Test Interface

The B11 provides IEEE 1149.1 and COP functions for facilitating board testing and chip debug. The IEEE 1149.1 test interface provides a means for boundary-scan testing of the PowerPC CPU and the board to which it is attached. The COP function shares the IEEE 1149.1 test port, provides a means for executing test routines, and facilitates chip and software debugging.

Connector types:

- Two 8-pin plugs, 2.54mm pitch, square pins ∅ 0.635mm gold
- Mating connector:
  Two 8-pin receptacles, high-precision, 2.54mm pitch, for square pins ∅ 0.635mm gold, 6.9mm height

Table 9. *Pin Assignment of the 16-pin IEEE 1149.1 Test Connector*



1	-	2	TDO
3	TRST	4	TDI
5	VCC	6	-
7	-	8	TCK
9	FLASHPROG	10	TMS
11	-	12	SRESET
13	-	14	HRESET
15	GND	16	-

3 MENMON

3.1 General

MENMON is an assembly-language debugger with a simple user console interface and can easily be extended and ported.

MENMON for B11 also uses some parts of Motorola's DINK32 and provides extensions for user interface, configuration, debugging and self test.

Purpose

- Debugging applications without any operating system
- Bootstrapping operating systems
- Hardware testing

Features

- Auto-configuration for PCI devices on the board and devices on secondary PCI buses
- Interrupt routing of all on-board devices and of all devices on secondary PCI buses
- SDRAM size detection, reading and checking (Serial Presence Detect Data Structure)
- Flash programming with password protection of MENMON spaces
- Primary/secondary MENMON
- Subset of Motorola PPCBug system calls implemented

3.2 Console

MENMON for B11 can communicate either through the COM1 port (available through the I/O connector) or through the first serial COM port of a P10 PC•MIP mezzanine module or AD45 mezzanine adapter.

Additionally, if a P1 graphics PC•MIP module is found on the PCI bus, any console output will also appear on the VGA display. The same is true if you connect a PS/2 keyboard to the I/O connecter. Characters can then be entered on the RS232 console and on the PS/2 keyboard.

The P10 or AD45 COM port is used only if

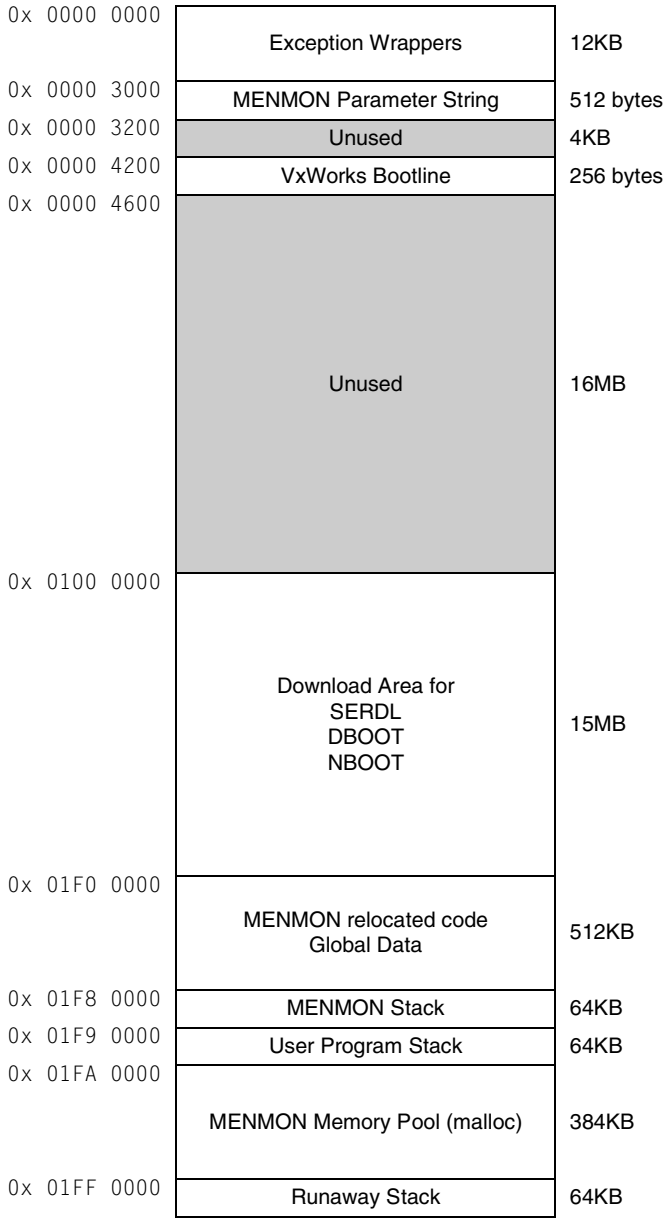
- a P10/AD45 is detected on the local PC•MIP slots and
- the hex switch on the B11 is set to between 8 and F. (N.B.: This function cannot be used on hardware revisions 00.xx.xx since the hex switch is not present there.)

Otherwise the COM1 port is used.

The default setting of the COM ports is 9600 baud, eight data bits, no parity, one stop bit.

3.3 B11 MENMON Memory Map

Figure 11. *MENMON Address Mapping*



3.4 MENMON Start-up

3.4.1 User LEDs

There are two LEDs available on the I/O connector (see Chapter 2.8 I/O Connector on page 27) or on the AD45 mezzanine adapter. The LEDs display the state of the boot like a counter.

The exact sequence of the LEDs, i.e. when each LED will light, depends on the MENMON version. If you have any problems during start-up, please turn to MEN's support at support@men.de and give your MENMON version.

3.4.2 Boot Sequence

The assembler part of MENMON initializes the CPU and the PCI bridge MPC107 integrated in the MPC8245 (memory interface), and the monitor will be relocated to the main memory.

All known devices will be initialized.

The primary MENMON looks for a valid secondary MENMON and starts it unless the ABORT button is pressed, which is available on the I/O connector. ("Valid" means the size is between 0x0000 and 0x80000 and the checksum is valid.)

If you press the ABORT button for more than five seconds, the MENMON settings in the EEPROM are restored with default values.

MENMON checks whether there is a valid "startup" string stored in EEPROM. If valid, all commands in the "startup" string are executed. (See Chapter 3.4.3 Configuring the MENMON Start-up Procedure on page 45.)

If no startup string was present, MENMON jumps to the operating system bootstrapper whose address can be configured using the EE-BS command.

The MENMON command line interface will appear if the ESC key is pressed or the bootstrapper address is set to an invalid address (i. e. 0x0 or 0xFFFFFFFF)

3.4.3 Configuring the MENMON Start-up Procedure

MENMON can be configured to automatically execute commands at start-up, for example to boot from disk. The EE-STARTUP command can be used to configure these commands. The EEPROM stores a string (max. 79 characters) that is comprised of commands that are executed at startup, e.g:

```
DBOOT 1 FILE=MYBOOT; NBOOT
```

MENMON performs these commands until one of the commands passes control to a loaded image.

The "EE-STARTUP -" command can be used to deactivate autoexecution of the string. When the string is inactive, MENMON calls its BO command at start-up.

3.4.4 Self Tests

At start-up the monitor runs self tests depending on the current self test level. (OFF, QUICK or EXTENDED). The MENMON behavior depends on the current stop on error mode (NO HOLD or HOLD).

Figure 12. *MENMON - Power On Self Test Output with Self Test Message Mode EXTENDED*

```
                    SYSTEM DIAGNOSE
                    ===============
                    === PCI ===
MPC107              DEV 0       ==> OK
ALI1543 PCI2ISA     DEV 12      ==> OK
ALI1543 IDE         DEV 1B      ==> OK
ALI1543 PMU         DEV 1C      ==> OK
VME Bridge          DEV 19      ==> OK
PCMIP I             DEV 1A      ==> NOT FOUND
PCMIP II            DEV 1D      ==> FOUND
                    === SMB ===
SO-SIMM SPD         ==> OK
RTC                 ==> OK
LM75                ==> OK
WDOG                ==> OK
                    === HEX ===
HEX-SW              ==> 0xA OK
                    === FLASH ===
CHECKSUM            ==> OK
```

3.4.4.1 Self Tests in Detail

RTC

The RTC test is non-destructive. It writes and compares the RTC NVRAM.

PCI

This test scans the PCI bus with configuration cycles for on-board PCI devices. PC•MIPs will only be displayed if the configuration access is successful.

SMB

This test performs read accesses to all on-board SMB devices.

HEX Switch

This test reads and displays the current hex switch position.

MENMON Flash Checksum

This test checks the checksum of the current MENMON (primary/secondary). The first long word of MENMON contains the size, the second long word contains the expected checksum. The test computes the checksum by XORing each long word of MENMON with the next one, except for the first two long words.

ABORT Button

This test checks pressing and releasing of the ABORT button to test port pin GPI 0 of the M1543 (cf. Chapter 4.3 Implementation of M1543 PCI-to-ISA Bridge on page 80).

The test is not performed during Power On Self Test.

This test does not check the ABORT interrupt.

CPU

This test enters and displays the B11 clock configuration.

The test is not performed during Power On Self Test.

An error is detected for unknown PLL configuration for the installed CPU type.

3.5 MENMON Boot Methods for Client Programs

MENMON supports different methods to load and start client programs like operating systems or their bootstappers:

- Disk boot
- Network boot
- Execution from Flash.

3.5.1 MENMON BIOS Devices

For disk and network Boot, MENMON supports several device tables. At the lowest level there is the **controller device**, an instantation of a controller driver. For example an IDE controller is a controller device. Each controller device is assigned a **Controller Logical Unit Number** (CLUN), to refer to the controller device. The controller device table is built only at startup of the CPU and is never changed at runtime.

On the next level there are high-level **devices**. For example, an IDE or SCSI hard disk would be called a device by the MENMON BIOS. Each device is assigned a **Device Logical Unit Number** (DLUN) that is unique for the controller. The MENMON device table is built dynamically on request (entries are added by the IOI or DBOOT command, for example).

The **IOI** command can be used to display the CLUNs and DLUNs known by MENMON. **IOIN** just displays the currently known devices while **IOI** will search for devices behind each controller.

Example

```
MenMon> IOIN

====== [ Controller Dev Table ] =========
CLUN Driver              param1     param2     param3     Handle
0x00 IDE                 0x000001F0 0x000003F6 0x00000000 0x00000000
0x01 IDE                 0x00000170 0x00000376 0x00000000 0x00000000
0x10 Etherboot           0x80002200 0xE0000000 0x00001A00 0x00000000

====== [ Device Table ] =========
CLUN DLUN Device                          Type          Handle
0x00 0x00 SunDisk SDCFB-20                IDE HD        0x000FFCA0
0x01 0x00 IBM DDRS-34560D                 IDE HD        0x000FFC50
```

3.5.1.1 Controller Devices (CLUNs)

On startup, MENMON searches for all known onboard controllers (CLUN 0x00..0x0F) and for any other PCI device that is supported by the MENMON drivers. If additional controllers are found on the PCI bus, they receive CLUNs ≥ 0x10.

Table 10. *MENMON - Assignment for B11 Controller Devices*

CLUN	Controller
0x00	Primary IDE controller in ALI
0x01	Secondary IDE controller in ALI
0x10..0FE	Any other controller found that is supported by the MENMON drivers

3.5.1.2 High Level Devices (DLUNs)

Depending on the bus type, the DLUN is assigned differently:

Device LUNs (8-bit value)

For **IDE** devices:

7..0
0 = Master 1 = Slave

For **SCSI** devices:

7..4	3..0
SCSI ID	SCSI LUN (normally 0)

Example: A SCSI hard disk with ID 6 would have a DLUN of 0x60.

3.5.2 Disk Boot

Disk boot supports the following:

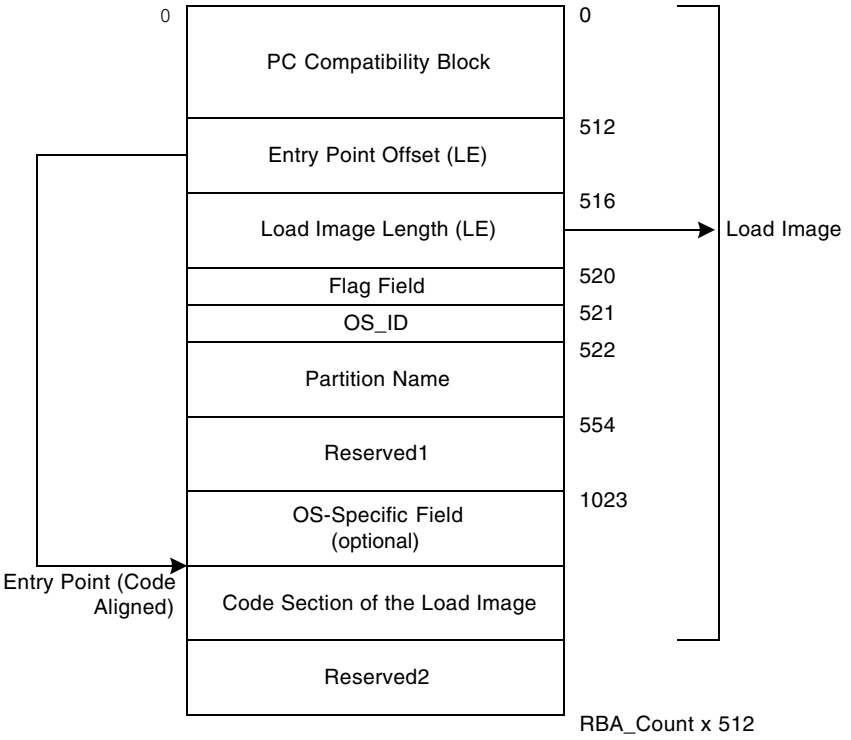
- Boot from any disk-like device: SCSI hard and floppy disks[1], IDE hard disks or CompactFlash.
- Supports PReP and DOS disk partitions as well as unpartitioned media.
- Supported file formats: ELF, PReP and PPCBOOT images.

To be able to boot from disk media, each medium must be prepared in the following way:

Partitions

Hard disks can have a partition table. MENMON supports the four partition entries in the first sector of the medium. The partition type must be either DOS (Type 0x01, 0x04, 0x06) or PReP (Type 0x41).

Figure 13. *MENMON - Layout of the 0x41-Type Partition (PReP)*



File System

With DOS-formatted partitions (or unpartitioned media) the file system must be a DOS FAT file system (12-bit or 16-bit FAT entries).

PReP (Type 0x41) partitions have no file system, the entire partition is viewed as a single file (no file name is required).

PReP partitions can contain either a PReP file (as in the above figure) or a PPCBOOT image.

[1] The B11 supports SCSI devices only through use of a PC•MIP module!

3.5.2.1 DBOOT Algorithm

The DBOOT command tries to find a bootable partition or file on any disk. If no parameters are specified, DBOOT will search for devices behind each known CLUN. On each disk found, it will check if there is a partition table on it, and checks with each partition if it is bootable or not.

Any PReP partition found is assumed to be bootable.

For DOS partitions, DBOOT searches if the DOS file system contains the specified file. The file name to be searched for can be configured in the EEPROM using the EE-BOOTFILE (or EE-VXBLINE) command. Only the file-name part of that name is used (e. g. if you configure EE-BOOTFILE /ata0/vxworks, then DBOOT looks for "vxworks").

The file name can also be passed to the command line to DBOOT (e. g. *DBOOT file=myboot*). The boot file must be in the root directory of the hard disk.

If no file name is configured in EEPROM and no file-name argument is passed to DBOOT, the filename defaults to "BOOTFILE".

3.5.2.2 Loading the Boot File

Once a bootable device/partition has been found, the DBOOT command starts to load the file. Regardless of the file format, the entire boot file will be loaded to MENMON's **download area** (0x01000000). (This address can be overridden using the LOAD parameter.) The load address **must not** be between 0x01F00000 and 0x01FFFFFF.

3.5.2.3 Starting the Loaded Program

RAW and PReP files will be executed at the load address.

For **RAW** files, the entry point, relative to the load address, can be specified through the START parameter to the DBOOT command. (The default start offset is 0, i.e. the program execution begins at the load address.)

PReP files begin with a header, which contains the entry point of the program. The START parameter will be ignored in this case.

ELF files will not be executed at the load address. Instead MENMON analyzes the ELF program header and sections, and the program sections will be relocated as specified in the ELF file. Here, the relocation address may be any address in RAM except the **runaway stack** and the load image itself. Only the physical address entries in the ELF program headers are used, virtual addresses are treated as physical addresses if the physical address entry is 0xFFFFFFFF.

Client Program Calling Conventions (for ELF, RAW and PReP files)

- Interrupts are disabled (MSR.EE is cleared).
- CPU is in Big Endian Mode.
- MMU is enabled. BATs are set up.
- Instruction Cache is enabled.
- R1 is set to the top of runaway stack - 512 bytes.
- R3 is set to 0 (no residual data available).
- R4 is set to the image loading address. (Not the relocation address!)
- R5..R7 are cleared.

3.5.2.4 PPCBOOT Images

MENMON additionally supports "ppcboot" compatible boot images which are often used to boot Linux. Those boot images can be generated using the *mkimage* tool.

PPCBOOT images have a CRC and can be compressed.

Although PPCBOOT images are typically used to boot Linux, they can be used for other operating systems as well.

MENMON first analyzes and decompresses (or moves) the loaded image as specified in the PPCBOOT header. If it detects that it's not a Linux kernel, it analyzes again whether the decompressed image is ELF, RAW or PReP format (see above).

If the file to be booted is a Linux kernel, then MENMON additionally provides support for an initialized RAMdisk (*initrd*). To boot the Linux kernel with an initialized RAMdisk, you must create a multi-file PPCBOOT image, where the first part is the Linux kernel and the second part is the *initrd* image. MENMON will move the *initrd* image to a 4K aligned boundary and passes the address to the Linux kernel.

When MENMON calls the Linux kernel, the registers have the following content:

- MMU is enabled.
- Instruction cache enabled.
- Data cache disabled.
- Stack is set to the normal MENMON stack.
- R1: Normal MENMON stack
- R3: Points to an array of boot info records (see below)
- R4: Start of initial RAMdisk (0 if none)
- R5: End of initial RAMdisk+1
- R6: Start of kernel command line
- R7: End of kernel command line+1

The bootinfo records have been introduced in recent 2.4.x Linux kernels and provide an alternative way to pass parameters from the bootloader to the Linux kernel.

MENMON passes the following boot info tags:

- ***BI_FIRST***–Start of Bootinfo records.
- ***BI_CMD_LINE***–Contains the kernel command line. This tag is missing when no command line is supplied.
- ***BI_INITRD***–Contains info about the initial RAMdisk. This tag is missing when no *initrd* has been included.
- ***BI_MENMON_PARAMETERS*** (0x1100)–Contains a copy of the MENMON parameter string that is normally located at address 0x3000.
- ***BI_LAST***–End of bootinfo records

Passing the Command Line to Linux Kernel

You can use the EEPROM to store the kernel command line. Use the *EE-KERPAR* command for this purpose. However, this can be overridden by adding parameter *KERPAR=* to NBOOT or DBOOT commands. In this case, the EEPROM command line will be ignored.

If neither the EEPROM command line nor the *KERPAR=* switch is present, no command line will be passed to the Linux kernel, and the Linux kernel will use the command line specified during kernel compilation.

3.5.2.5 Using the DBOOT Command

Syntax	DBOOT [*clun*] [*dlun*] [PART=*part*] [FILE=*file*] [LOAD=*addr*] [START=*off*] [HALT=*n*] [KERPAR=p1=x p2=y]	
Parameters	*clun*	Controller logical unit. If missing, DBOOT loops through all known controllers.
	dlun	Device logical unit. If missing, DBOOT automatically searches for devices.
	PART	Partition number [1..4]. If missing, DBOOT loops through all partitions.
	FILE	File name. Used when booting from a DOS FAT file system. The file must be present in the file system's root directory. If FILE is missing, the name "BOOTFILE" is used. The file name is ignored when booting from Type41 partitions.
	LOAD	Specifies the load address. This is the address where the entire image of the file is first loaded, regardless of the file format. If not specified, the *download area* is used.
	START	Specifies the entry point of the loaded program relative to its load address. Only used for RAW files. If START is not present, the entry point is equal to the load address.
	HALT	If this parameter is '1', MENMON is called again when the boot file was loaded. If this parameter is '2', MENMON is called when the load image was relocated, right after the first instruction of the program was executed.
	KERPAR	Parameters to add to kernel command line (only used when booting PPCBOOT image)

Examples

- Load PReP boot from second partition of CompactFlash:

```
DBOOT 0 0 PART=2
```

- Load file *MYBOOT* from IDE hard disk on secondary IDE channel, master:

```
DBOOT 1 1 FILE=MYBOOT
```

- Try to find a bootable device on secondary IDE:

```
DBOOT 1
```

- Boot Linux from PPCBOOT image and pass kernel parameters:

```
DBOOT FILE=busybox.img KERPAR='root=ramfs console=ttyS0,9600'
```

- Boot VxWorks from ATA:

```
MenMon> ee-vxbline

'.' = clear field;  '-' = go to previous field;  ^D = quit

boot device           :ata=0,0
processor number      :0
host name             :host
file name             :/ata0/vxworks
inet on ethernet (e)  :192.1.1.28
inet on backplane (b) :
host inet (h)         :192.1.1.22
gateway inet (g)      :
user (u)              :
ftp password (pw) (blank = use rsh):
flags (f)             :0x0
target name (tn)      :
startup script (s)    :
other (o)             :
Updating EEPROM..

MenMon> DBOOT 0
```

Hints

- Use the LS command to display the partition table and files on the device.
- In case of problems you can try to read raw sectors from disk using the DSKRD command.
- Use the EE-STARTUP command to perform the DBOOT command automatically at startup.

3.5.3 Network Boot

Note: You can use network boot only through a PC•MIP module!
(Currently, P12 is supported.)

Network boot supports the following:

- Boot a file using BOOTP and TFTP protocols via Ethernet.
- Boot a file using TFTP only (without BOOTP).
- Supported file formats: RAW, ELF and PReP.

This boot method requires a host computer running the TCP/IP daemons *tftpd* and optionally *bootp*. If you intend to boot via BOOTP, the host computer must also set up a table (usually called *bootptab*) containing an entry for each target system to be booted.

An entry in *bootptab* for B11 could look like this:

```
mysystem:sm=255.255.255.0:\
 hd=/usr/TFTPBOOT:\
 bs:ht=ether:vm=rfc1048:\
 ha=00c03a080003:\
 ip=192.1.1.25:\
 bf=mybootfile
```

At start-up, MENMON searches for the first available (and supported) Ethernet controller in the system. When the NBOOT command is issued, MENMON uses that controller (unless the CLUN parameter is specified) to send its BOOTP broadcast. The BOOTP server will respond with a packet containing the target's IP address, home directory and boot file. Now MENMON will fetch the specified file using the TFTP protocol.

However you can also boot through TFTP only. In this case, you must configure some parameters in the EEPROM. These parameters can be configured using either EE-VXBLINE or the EE-NETxxx parameters.

Example of Booting a Specified File

```
MenMon> ee-netip 192.1.1.28
MenMon> ee-nethost 192.1.1.22
MenMon>ee-bootfile /FWARE/PPC/MENMON/PORTS/B11/BIN/menmon.rom
MenMon> nboot tftp
Probing...[Tulip] Tulip 00:C0:3A:08:00:17 at membase = 0xF0001000
Performing ethernet autonegotiation (V2)...100BaseTx FD
Etherboot/32 version 4.2.5b for [Tulip]

My IP 192.1.1.28, Netmask=0xFFFFFF00 Server IP 192.1.1.22, GW IP
0.0.0.0
Loading /FWARE/PPC/MENMON/PORTS/B11/BIN/menmon.rom...
to 0x01000000        352 kB
Loaded 0x000580DC bytes
Starting RAW-file
```

As with the DBOOT command, the entire boot file will be loaded to MENMON's *download area* if not otherwise specified. Once the boot file has been loaded, the file is interpreted, relocated and executed in the same way as described for the DBOOT command. (See Chapter 3.5.2.3 Starting the Loaded Program on page 51.)

Client Program Calling Conventions

See Chapter Client Program Calling Conventions (for ELF, RAW and PReP files) on page 51.

3.5.3.1 Using the NBOOT Command

Syntax	NBOOT [BOOTP=*??*] [TFTP=*??*] [CLUN=*clun*] [FILE=*file*] [LOAD=*addr*] [START=*addr*] [HALT=*n*] [KERPAR=p1=x p2=y]	
Parameters	*BOOTP*	(Default) Obtain IP address from BOOTP server. Then boot via TFTP.
	TFTP	Use TFTP method only. Use parameters specified by EE-NETxx commands.
	CLUN	Specifies the controller that should be used for network boot. If CLUN is not present, the first available controller is used.
	FILE	File name to be sent within the BOOTP request. If FILE is not present, the file name must be provided by the BOOTP server (using the "bf" tag). A file name from the BOOTP server always takes precedence.
	LOAD	See Chapter 3.5.2.5 Using the DBOOT Command on page 53
	START	See Chapter 3.5.2.5 Using the DBOOT Command on page 53
	HALT	See Chapter 3.5.2.5 Using the DBOOT Command on page 53
	KERPAR	See Chapter 3.5.2.5 Using the DBOOT Command on page 53

3.5.3.2 Ethernet Medium Selection

MENMON currently supports Ethernet controllers using the DEC21xxx chips. These chips can be on the P12 PC•MIP mezzanine and on some other PCI hardware.

The medium to be used (10Mbit or 100Mbit, full duplex or half duplex) is stored in the SROM that is connected to the DEC chip. Normally, "Autoselect" is stored here, so MENMON will attempt to select the Ethernet medium automatically.

MENMON's DEC21MEDIA command can be used to display or to change the current medium selection.

3.6 Updating Flash Devices

MENMON provides the possibility of updating Flash and disk devices on the B11 via the serial console interface or via Ethernet.

3.6.1 Download via Serial Interface

In order to program Flash or disk devices, you need to send a file from a host computer to the target. On the host computer, you need a terminal emulation program such as HyperTerm or Minicom.

The download file name extension determines the destination device and the offset within that device. For example, a file named *myfile.f00* will be programmed into Flash sector 0.

Table 11. *MENMON - Download Destination Devices*

Device Abbreviation	Flash Device	Sector Size
F	Flash	See Table 12, MENMON - Flash Sectors, on page 58
E	Serial EEPROM[1]	1 byte
D	SDRAM	2 bytes
C	IDE (CompactFlash)	512 bytes
S	SCSI ID0	Sector size from drive

[1] If you want to program the EEPROM and use the file extension to specify the start address, note that the highest start address you can state is 0x63 (with extension *.E99*).

Two special extensions are available for MENMON update:

- *xxx.PMM* is an alias for .F16 and updates the primary MENMON.
- *xxx.SMM* is an alias for .F24 and updates the secondary MENMON.

When a file is larger than one sector, the following sector of the device will also be programmed.

The update file is transferred to DRAM before being programmed to Flash. The DRAM of the B11 must therefore be large enough for the entire download file. The update file may be max. 15MB.

Table 12. *MENMON - Flash Sectors*

Flash Sector	Address
0	0x000000
1	0x010000
2	0x020000
3	0x030000
4	0x040000
5	0x050000
6	0x060000
7	0x070000
8	0x080000
9	0x090000
10	0x0A0000
11	0x0B0000
12	0x0C0000
13	0x0D0000
14	0x0E0000
15	0x0F0000
16	0x100000
17	0x110000

Flash Sector	Address
18	0x120000
19	0x130000
20	0x140000
21	0x150000
22	0x160000
23	0x170000
24	0x180000
25	0x190000
26	0x1A0000
27	0x1B0000
28	0x1C0000
29	0x1D0000
30	0x1E0000
31	0x1F0000
32	0x1F8000
33	0x1FA000
34	0x1FC000

3.6.2 Performing the Download

You must connect your host to B11's COM1 interface.

Before you start the download, change the MENMON console baudrate to 115,200 baud (enter *EE-BAUD 115200* and reset B11).

To start download enter *SERDL* in the MENMON command line. You must specifiy a password if you want to update the primary MENMON, secondary MENMON or disk devices:

- *SERDL PMENMON* for primary MENMON
- *SERDL MENMON* for secondary MENMON
- *SERDL DISK* for disk devices

3.6.3 Update from Disk or Network

It is also possible to program Flash with a file from a disk or network:

☑ Load the file into memory:

```
DBOOT HALT=1
or
NBOOT HALT=1
```

☑ Program the Flash (in this case OS bootstrapper):

```
PFLASH F 0 100000
```

This programs the first Mbyte of Flash.

3.7 MENMON User Interface

3.7.1 Command Line Editing

MENMON provides a rudimentary command line editor:

<CTRL> <H>	Backspace and delete previous character
<CTRL> <X>	Delete whole line
<CTRL> <A>	Retrieve last line

3.7.2 Numerical Arguments

Most MENMON commands require one or more arguments. Numerical arguments may be numbers or simple expressions:

<num>	*num* is interpreted as a hexadecimal value
$<num>	Same as above
#<num>	*num* is interpreted as a decimal value
%<num>	*num* is interpreted as a binary value
.<REG>	Use the value of register <REG>

These arguments can be combined using the arithmetic operators "+" and "-".

Example:[1]

`MenMon> D 10000`	Dumps address `0x10000`

[1] Some of the addresses used in our examples may not be suitable for your board's address mapping. If you want to try out MENMON's functions, please compare the example addresses with your mapping first!

3.7.3 MENMON Command Overview

Table 13. *MENMON Command Overview*

Command	Description
H	Print help
IOI	Scan for BIOS devices
NBOOT [<opts>]	Boot from network
DEC21MEDIA <clun> <med>	Select Ethernet medium
DBOOT [<clun>] [<dlun>] [<opts>]	Boot from disk
LS <clun> <dlun> [<opts>]	List files/partitions on device
DSKRD <args>	Read blocks from RAW disk
DSKWR <args>	Write blocks to RAW disk
BIOS_DBG <mask>	Set MMBIOS debug level
I [<D>]	List B11 information
EE[-xxx] [<arg>]	Serial EEPROM commands
DIAG [<arg>]	System diagnosis
SERDL [<passwd>]	Update Flash using YModem protocol
ERASE <D> [<O>] [<S>]	Erase Flash sectors
PFLASH <D> <O> <S> [<A>]	Program Flash
AS <addr> [<cnt>]	Assemble memory
DI [<addr>] [<cnt>]	Disassemble memory
GO [<addr>]	Jump to user program
S[RFO-] [<addr>]	Single step
BO [<addr>]	Call OS bootstrapper
B[DC#] [<addr>]	Set/display/clear breakpoints
.C[RFM] name	CPU User Register Change
.[RFM?] [name]	CPU User Register Display
C[BWLN#] <expr>	Change memory
D [<addr>] [<cnt>]	Dump memory
FI <from> <to> <val>	Fill memory (byte)
MC <adr1> <adr2> <cnt>	Compare memory
MO <from> <to> <cnt>	Move (copy) memory
MS <from> <to> <val>	Search pattern in memory
MT[BWLFD] <from> <to>	Memory test
PCID[+] <devNo> [<busNo>]	PCI config register dump
PCIC <devNo> <addr> [<busNo>]	PCI config register change
PCIR	List PCI resources
PCI-VPD[-] <devNo> [<busNo>] [<capId>]	PCI Vital Product Data dump
PCI	PCI probe
WDOG[-xxx]	Watchdog commands

3.8 Board Setup

3.8.1 ALI 1543

The PCI-to-ISA southbridge contains preconfigured and unconfigured Plug and Play devices.

MENMON enables and configures the following devices:

- COM1
- COM2
- Keyboard
- Mouse
- Primary/secondary IDE
- DMA controller
- PMU
- SMB controller

MENMON disables the following devices:

- USB

3.8.2 PCI Auto-Configuration

MENMON maps all detected local PCI devices to PCI memory and PCI I/O space. PCI bus masters are enabled. PCI bus interrupts are routed and configured in configuration space.

The cacheline size and latency timer registers of all PCI devices are initialized:

PCI Latency Timer = 0x40 = 1.94µs

The information command I displays the current PCI configuration:

Figure 14. *MENMON - Example PCI Configuration*

```
MenMon> i d
B11 Info
=======
...
*PCI
busNo devNo funcNo DEV ID  VEN ID  MEM MAPPED CFG REGS at
===== ===== ====== ======  ======  ======================
0x 0  0x 0  0x 0   0x0003  0x1057  not avail
0x 0  0x12  0x 0   0x1533  0x10B9  0x80840000
0x 0  0x19  0x 0   0x5056  0x1172  not avail
0x 0  0x1B  0x 0   0x5229  0x10B9  not avail
0x 0  0x1C  0x 0   0x7101  0x10B9  not avail
0x 0  0x1D  0x 0   0x0019  0x1011  not avail

NUMBER OF MAPPED PCI BUSSES => 0
PCI IO:
    START => FE002200
    END   => FE00EFFF
    ALLOC => FE002280
PCI MEMORY:
    START => 80000000
    END   => 9EFFFFFF
    ALLOC => 82000400
PCI INT ROUTING:
    INTA =>   7
    INTB =>   9
    INTC =>  10
    INTD =>  11

PCI BRIDGES:
    PrimBus DevNo SecBus
    ------- ----- ------
```

There are two commands to control some features on the PCI bus.

- ***EE-PCI-STGATH*** controls PCI store gathering of CPU->PCI cycles.
- ***EE-PCI-SPECRD*** controls read prefetching of external master accesses to the system memory.

There are several commands available to show and modify PCI configuration:

- ***PCI*** scans the entire bus hierarchy and displays the device and vendor ID of each device found.
- ***PCIR*** shows the allocated PCI I/O and memory resources for each device.
- ***PCID*** shows the entire PCI configuration space of the specified device.
- ***PCIC*** allows you to change the values of any PCI config space register.
- ***PCI-VPD*** shows the "vital product data" on devices that support it.

3.8.3 SDRAM DIMM Configuration

The configuration EEPROM will be read over the System Managment Bus. The monitor software checks the configuration data during boot. The SDRAM controller is set up according to the information found in the serial presence detect (SPD) EEPROM. If no valid SPD can be found, defaults are used.

A bad SPD checksum is tolerated as long as the rest appears reasonable.

3.8.4 VMEbus Configuration

There are some configuration commands available for the VME interface. Please note that the CPU address mapping to access the VMEbus from the B11 does not need to be configured (see Chapter 4 Organization of the Board on page 74).

EE-VME-IRQ <mask> — Configuring the interrupt handler

This command controls which of the seven VME interrupt levels should be enabled and should be handled by the B11. Each bit in <mask> corresponds to one interrupt level (bit7 = level 7, bit1 = level 1).

For example *EE-VME-IRQ 0x30* would enable levels 5 and 4.

Bit 0 in <mask> controls enabling of the special ACFAIL interrupt.

Note: MENMON will never enable any interrupt in the VME bridge. The *EE-VME-IRQ* command is only here to provide a common way to enable interrupt levels for all operating systems. The operating system is responsible for reading the setting made by *EE-VME-IRQ* and for enabling the corresponding interrupts. The operating system has to read either the MENMON parameter string (cf. Chapter 3.10.1 Additional MENMON Parameters on page 73) or the EEPROM.

EE-VME-A24SA <val> -— Configuring the VME slave address

The VME address of the B11's shared SRAM can be configured through the *EE-VME-A24SA* command. <val> configures A23..A20 of the compare address, in steps of 1MB. For example *EE-VME-A24SA 8* configures a compare address of 0x800000.

Specifying *EE-VME-A24SA FF* disables access to the SRAM from VMEbus.

By default, the slave interface is disabled.

EE-VME-REQ <val> — Configuring the VME bus requester method

The VME bus requester can be configured to ROR (release on request) or RWD (release when done):

- *EE-VME-REQ 0* requester configured for ROR (default)
- *EE-VME-REQ 1* requester configured for RWD

3.8.5 Watchdog Configuration

By default, the B11 watchdog is disabled.

The watchdog can be enabled through *WDOG-TOUT <ms>* where <ms> specifies the watchdog timeout in milliseconds. Possible values are 0 (disable watchdog), 800, 1600, 3200, and 6400.

Once the watchdog is enabled, it must be served by toggling the ALI GPIO2 pin. If the software fails to toggle this pin in time, the CPU is reset.

MENMON automatically and continuously serves the watchdog until the operating system is started.

3.8.6 Hex Switch

The hex switch is completely user-configurable. With MENMON it has only one function: at hex position "0" or "8" there will be a delay after each initialization step, so that the boot procedure is slowed down. This function is provided for diagnostic purposes. For normal operation of the board, you should set the hex switch to a position between "1" and "F".

If the hex switch is set to 8 and F, the console can be redirected to a P10 PC•MIP or AD45 adapter. See Chapter 3.2 Console on page 43.

Table 14. *Hex-Switch Settings*

Setting	Description
0	User-defined, but delay after each initialization step
1..F	User-defined, no additional delay during boot

3.9 MENMON System Calls

This chapter describes the MENMON System Call handler, which allows system calls from user programs. MENMON implements a small subset of the system calls implemented in Motorola's PPCBug. The implemented system calls are binary-compatible with PPCBug.

The system calls can be used to access selected functional routines contained within the debugger, including input and output routines. The System Call handler may also be used to transfer control to the debugger at the end of a user program.

3.9.1 Invoking System Calls

The System Call handler is accessible through the *SC* (system call) instruction, with exception vector 0x00C00 (System Call Exception). To invoke a system call from a user program, insert the following code into the source program. The code corresponding to the particular system routine is specified in register R10. Parameters are passed and returned in registers R3 to *R*n, where *n* is less than10.

```
ADDI R10,R0,$XXXX
SC
```

$XXXX is the 16-bit code for the system call routine, and *SC* is the system call instruction (system call to the debugger). Register R10 is set to 0x0000*XXXX*.

3.9.2 System Calls

3.9.2.1 *BRD_ID*

Name	BRD_ID — Return pointer to board ID packet
Code	$0070
Description	This routine returns a pointer in R03 to the board identification packet. The packet is built at initialization time. The format of the board identification packet is shown below. MENMON only implements some fields of the original PPCBug system call.

Table 15. *MENMON System Calls - BRD_ID Fields*

	31	24	23	16	15	8	7	0
0x00	Eye Catcher							
0x04	reserved							
0x08	Packet Size				reserved			
0x0C	reserved							
0x10	reserved							
0x14	CLUN				DLUN			
0x18	reserved							
0x1C	reserved							

Eye Catcher	Word containing ASCII string "BDID"
Packet Size	Half-word containing the size of the packet
CLUN	Logical Unit Number for the boot device controller
DLUN	Logical Unit Number for the boot device

Entry Conditions	-	
Exit Conditions different from Entry	R03: Address (word)	Starting address of ID packet

Note: *CLUN* and *DLUN* are initialized according to the device that was last booted (for example, DBOOT or NBOOT command).

3.9.2.2 *OUT_CHR*

Name	*OUT_CHR* — Output character routine
Code	$0020
Description	This routine outputs a character to the default output port.
Entry Conditions	R03: Bits 7 through 0 Character (byte)
Exit Conditions different from Entry	Character is sent to the default I/O port.

3.9.2.3 *IN_CHR*

Name	*IN_CHR* — Input character routine
Code	$0000
Description	*IN_CHR* reads a character from the default input port. The character is returned in the LSB of R03.
Entry Conditions	-
Exit Conditions different from Entry	R03: Bits 7 through 0 contain the character returned R03: Bits 31 through 8 are zero.

3.9.2.4 *IN_STAT*

Name	*IN_STAT* — Input serial port status routine
Code	$0001
Description	*IN_STAT* is used to see if there are characters in the default input port buffer. R03 is set to indicate the result of the operation.
Entry Conditions	No arguments required
Exit Conditions different from Entry	R03: Bit 3 (ne) = 1; Bit 2 (eq) = 0 if the receiver buffer is not empty. R03: Bit 3 (ne) = 0; Bit 2 (eq) = 1 if the receiver buffer is empty.

3.9.2.5 *RTC_RD*

Name	*RTC_RD* — Read the RTC registers
Code	$0053
Description	RTC_RD is used to read the Real-Time Clock registers. The data returned is in packed BCD. The order of the data in the buffer is: Begin buffer — Buffer + eight bytes YY — Year (2 nibbles packed BCD) MM — Month (2 nibbles packed BCD) (1..12) DD — Day of month (2 nibbles packed BCD) (1..31) dd — Always 0 H — Hour (2 nibbles packed BCD) (0..23) M — Minutes (2 nibbles packed BCD) (0..59) S — Seconds (2 nibbles packed BCD) (0..59)
Entry Conditions	R03: Buffer address where RTC data is to be returned
Exit Conditions different from Entry	Buffer now contains date and time in packed BCD format.

3.9.2.6 *DSK_RD*

Name	*DSK_RD* — Disk read routine
Code	$0010
Description	This routine is used to read blocks of data from the specified disk device. Information about the data transfer is passed in a command packet which has been built somewhere in memory. (The user program must first manually prepare the packet.) The address of the packet is passed as an argument to the routine. The command packet is eight half-words in length and is arranged as follows:

Table 16. *MENMON System Calls - DSK_RD Fields*

	15 — 8	7 — 0
Ø x ØØ	CLUN	DLUN
Ø x Ø2	Status Half-Word	
Ø x Ø4	Memory Address	Most Significant Half-Word
Ø x Ø6		Least Significant Half-Word
Ø x Ø8	Block Number (Disk)	Most Significant Half-Word
Ø x ØA		Least Significant Half-Word
Ø x ØC	Number of Blocks	
Ø x ØE	Flag Byte	Address Modifier

CLUN	Logical Unit Number (LUN) of controller to use
DLUN	Logical Unit Number (LUN) of device to use
Status	This status half-word reflects the result of the operation. It is zero if the command completed without errors.
Memory Address	Address of buffer in memory. Data is written starting at this address.
Block Number	For disk devices, this is the block number where the transfer starts. Data is read starting at this block.
Number of Blocks	The number of blocks to read from the disk. For streaming tape devices, the actual number of blocks transferred is returned in this field.
Flag Byte	Not implemented by MENMON
Address Modifier	Not used

Entry Conditions	R03: 32-bit address of command packet

Exit Conditions different from Entry	Status half-word of command packet is updated. Data is written into memory. R03: Bit 3 (ne) = 1; Bit 2 (eq) = 0 if errors. R03: Bit 3 (ne) = 0; Bit 2 (eq) = 1 if no errors.

Note: MENMON's internal status codes are returned in *Status*.

3.10 VxWorks Bootline

MENMON passes a string to the client program that confirms to the standard VxWorks bootline. This string is copied to a fixed address before the client program is called.

MENMON stores the VxWorks bootline in the serial EEPROM. MENMON command EE-VXBLINE allows you to change the bootline interactively (same behavior as VxWorks *bootChange()* routine).

There are alternative commands to modify only specific parameters within the bootline.

The parameters in the bootline are used both by MENMON and by operating system bootstrappers.

The address of the bootline string is `0x4200` on all PowerPC platforms and has space for 256 characters.

The bootline has the following form:

```
bootdev(unitnum,procnum)hostname:filename e=# b=# h=# g=# u=userid
pw=passwd f=#
tn=targetname s=startupscript o=other
```

The bootline is a null-terminated ASCII string. Example:

```
enp(0,0)host:/usr/wpwr/target/config/mz7122/vxWorks e=90.0.0.2
b=91.0.0.2 h=100.0.0.4    g=90.0.0.3 u=bob pw=realtime f=2 tn=target
s=host:/usr/bob/startup o=any_string
```

Table 17. *MENMON VxWorks Bootline - List of Parameters and their Usage*

Parameter	Description	Special Command	Used by MENMON
boot device + unit number	Device name of boot device		No
processor number			No
host name	Name of host to boot from		No
file name	File name of file to be booted	EE-BOOTFILE	Yes, for NBOOT and DBOOT
inet on ethernet (e=)	IP address and optional subnet mask of this machine on Ethernet (e. g. 192.1.1.28:ffffff00)	EE-NETIP	Yes, for NBOOT
inet on backplane (b=)	IP address on backplane		No
host inet (h=)	IP address of host to boot from	EE-NETHOST	Yes, for NBOOT
gateway inet (g=)	IP address of gateway	EE-NETGW	Yes, for NBOOT
user (u=)	User name		No
ftp password (pw=)	Password		No
flags (f=)	Flags for VxWorks		No
target name (tn=)	Name of this machine	EE-NETNAME	No

Parameter	Description	Special Command	Used by MENMON
startup script (s=)	Startup script for VxWorks	EE-KERPAR	Yes, when booting PPCBOOT images containing a Linux kernel
other (o=)	Other devices to initialize in VxWorks		No

3.10.1 Additional MENMON Parameters

Client programs often need to query certain parameters which are already set up or detected by MENMON. In the past, client programs had to read the EEPROM or access some registers directly in order to get these parameters.

The new method allows MENMON to pass certain parameters to the client program. These parameters are stored in an separate ASCII string. The advantages lie in common access to these parameters over the range of PPC boards and saving time to boot.

The address of the parameter string is `0x3000` on all PowerPC platforms and has space for 512 characters.

Table 18. *MENMON - Common Parameters Passed by All MENMONs*

Parameter	Description
MPAR	Magic word at beginning of string
brd=B011	Product name of the board
brdrev=*xx.yy.zz*	Board revision
brdmod=*xx*	Board model
sernbr=*xxxx*	Serial number (decimal)
cbr=*baud*	Console baud rate in bits/s (decimal)
cons=*dev*	Selected console as an ASCII string (“COM1” or “P10” or "VGA", if both a graphics module and a PS/2 keyboard were found)
mem0=*size*	Size of main memory in kbyte (decimal)
cpu=*name*	CPU type (MPC8240, MPC8245)
cpuclk=*f*	CPU frequency in MHz (decimal)
memclk=*f*	Memory bus frequency in MHz (decimal)
vmeirqenb=*mask*	Enabled interrupt levels as defined by *EE-VME-IRQ*

Example

```
00003000:  4D504152 20627264  3D423031 31206272  MPAR brd=B011 br
00003010:  64726576 3D30302E  30302E30 30206272  drev=00.00.00 br
00003020:  646D6F64 3D303020  7365726E 62723D32  dmod=00 sernbr=2
00003030:  20636272 3D313135  32303020 636F6E73   cbr=115200 cons
00003040:  3D434F4D 31206370  753D4D50 43383234  =COM1 cpu=MPC824
00003050:  30206370 75636C6B  3D323530 206D656D  5 cpuclk=300 mem
00003060:  636C6B3D 31303020  766D6569 7271656E  clk=100 vmeirqen
00003070:  623D4645 00084028  21142201 40001020  b=FE..@(!."."@..
```

4 Organization of the Board

To install software on the B11 board or to develop low-level software it is essential to be familiar with the board's address and interrupt organization.

4.1 Memory Mappings

The memory mapping of the B11 complies with the CHRP (Common Hardware Reference Platform) Specification. The MPC107 host bridge is set to map B to support this mapping.

4.1.1 Processor View of the Memory Map

Table 19. *Memory Map - Processor View*

CPU Address Range	Size	Description
0x 0000 0000 .. 0FFF FFFF	1GB	DRAM
0x 1000 0000 .. 7FFF FFFF	1.8GB	Reserved
0x 8000 0000 .. FCFF FFFF	2GB-48MB	PCI Memory Space
0x FD00 0000 .. FDFF FFFF	16MB	PCI ISA Memory Space
0x FE00 0000 .. FE00 FFFF	64KB	PCI ISA I/O Space
0x FE80 0000 .. FEBF FFFF	4MB	PCI I/O Space (not used)
0x FEC0 0000 .. FEDF FFFF	2MB	PCI Config Addr. Reg.
0x FEE0 0000 .. FEEF FFFF	1MB	PCI Config Data. Reg.
0x FEF0 0000 .. FEFF FFFF	1MB	PCI IACK Space
0x FFE0 0000 .. FFFF FFFF	2MB	Boot Flash (8-bit)

Table 20. *Address Mapping for PCI*

Address Range	Description
PCI Memory Space (addresses as seen on PCI bus)	
0x 8000 0000 .. 81FF FFFF	VME bridge (see VME mapping)
0x 8200 1000 .. 9EFF FFFF	Available for PCI auto configuration
0x 9F00 0000 .. FFF FFFF	MPC8245 embedded utility block
PCI I/O Space (addresses as seen on PCI bus)	
0x 0000 .. 21FF	Fixed addresses of ISA devices (see Chapter 4.1.2 PCI/ISA I/O Space Memory Map on page 76)
0x 2200 .. EFFF	Available for PCI I/O space auto-configuration
0x F000 .. FFFF	ALI IDE bus mastering

Table 21. *BATS set up by MENMON*[1]

Addr	BAT	Description
0x F000 0000 .. FFFF FFFF	0	PCI ISA & I/O & IACK and boot Flash IBAT: Caching enabled
0x 0000 0000 .. *xx*00 0000 (depending on DRAM configuration)	1	DRAM IBAT: Caching enabled
0x 8000 0000 .. 8FFF FFFF	2	PCI/ISA I/O Space
0x 9000 0000 .. 9FFF FFFF	3	PCI Memory Space

[1] Unless otherwise stated, all BATS are initialized with W I M !G.

4.1.2 PCI/ISA I/O Space Memory Map

This memory map complies to the ISA I/O address assignments. Refer to data sheet "ALADDIN M1543: Desktop South Bridge, version 1.25, Jan. 1998" for configuration registers.

Table 22. *PCI/ISA I/O Space Memory Map (addresses as seen from CPU)*

CPU Address Range	Device	Register
0x FE00 0000 .. FE00 000F	M1543	DMA1 (slave)
0x FE00 0020	M1543	INT_1 (master) Control Register
0x FE00 0021	M1543	INT_1 (master) Mask Register
0x FE00 0040	M1543	Timer Counter - Channel 0 Count
0x FE00 0041	M1543	Timer Counter - Channel 1 Count
0x FE00 0042	M1543	Timer Counter - Channel 2 Count
0x FE00 0043	M1543	Timer Counter Command Mode Register
0x FE00 0060	M1543	Read_access Clear IRQ[12] (for PS2), IRQ[1] Latched Status
0x FE00 0060	M1543	Keyboard Data Buffer
0x FE00 0061	M1543	NMI and Speaker Status and Control
0x FE00 0064	M1543	Keyboard Status(R)/Command(W)
0x FE00 0080 .. FE00 009F	M1543	DMA Channel x Page Register
0x FE00 00A0	M1543	INT_2 (slave) Control Register
0x FE00 00A1	M1543	INT_2 (slave) Mask Register
0x FE00 00C0 .. FE00 00DF	M1543	DMA2 (master)
0x FE00 00F0	M1543	Coprocessor Error Ignored Register
0x FE00 0170 .. FE00 0177	M1543	IDE Secondary registers part A
0x FE00 01F0 .. FE00 01F7	M1543	IDE Primary registers part A
0x FE00 02F8 .. FE00 02FF	M1543 Super I/O	UART2 controller
0x FE00 0378 .. FE00 037F	M1543 Super I/O	Parallel Port Controller
0x FE00 03F0	M1543 Super I/O	Config Port Index
0x FE00 03F1	M1543 Super I/O	Config Port Data
0x FE00 0376 .. FE00 0377	M1543	IDE Secondary registers part B
0x FE00 03F6 .. FE00 03F7	M1543	IDE Primary registers part B
0x FE00 03F8 .. FE00 03FF	M1543 Super I/O	UART1 controller
0x FE00 040B	M1543	DMA1 Extended Mode Register
0x FE00 0481 .. FE00 048B	M1543	DMA High Page Registers
0x FE00 04D0	M1543	INT_1 (master) Edge/Level Control
0x FE00 04D1	M1543	INT_2 (slave) Edge/Level Control
0x FE00 04D6	M1543	DMA2 Extended Mode Register
0x FE00 1800 .. FE00 181E	M1543	SMB Controller
0x FE00 2000 .. FE00 201F	M1543	PMU of ALI
0x FE00 F000 .. FE00 F00F	M1543	IDE bus master registers

4.1.3 VMEbus Memory Mapping

MENMON maps BAR0 of the PCI to VME bridge always at 0x 8000 0000, so the following addresses will always be valid:

Table 23. *VMEbus Memory Mapping*

Size	Mapped by MENMON to	Function
16MB	0x 8000 0000	VME A24 (standard) space
64KB	0x 8100 0000	VME A16 (short) space
1MB	0x 8140 0000	Local SRAM
64 bytes	0x 8180 0000	VME Bridge Control Registers
16 bytes	0x 81C0 0000	VME IACK space

4.2 Interrupt Handling

The B11 supports both maskable and nonmaskable interrupts. The interrupt controller is located inside the M1543 PCI-to-ISA bridge.

Figure 15. *B11 Interrupt Structure*

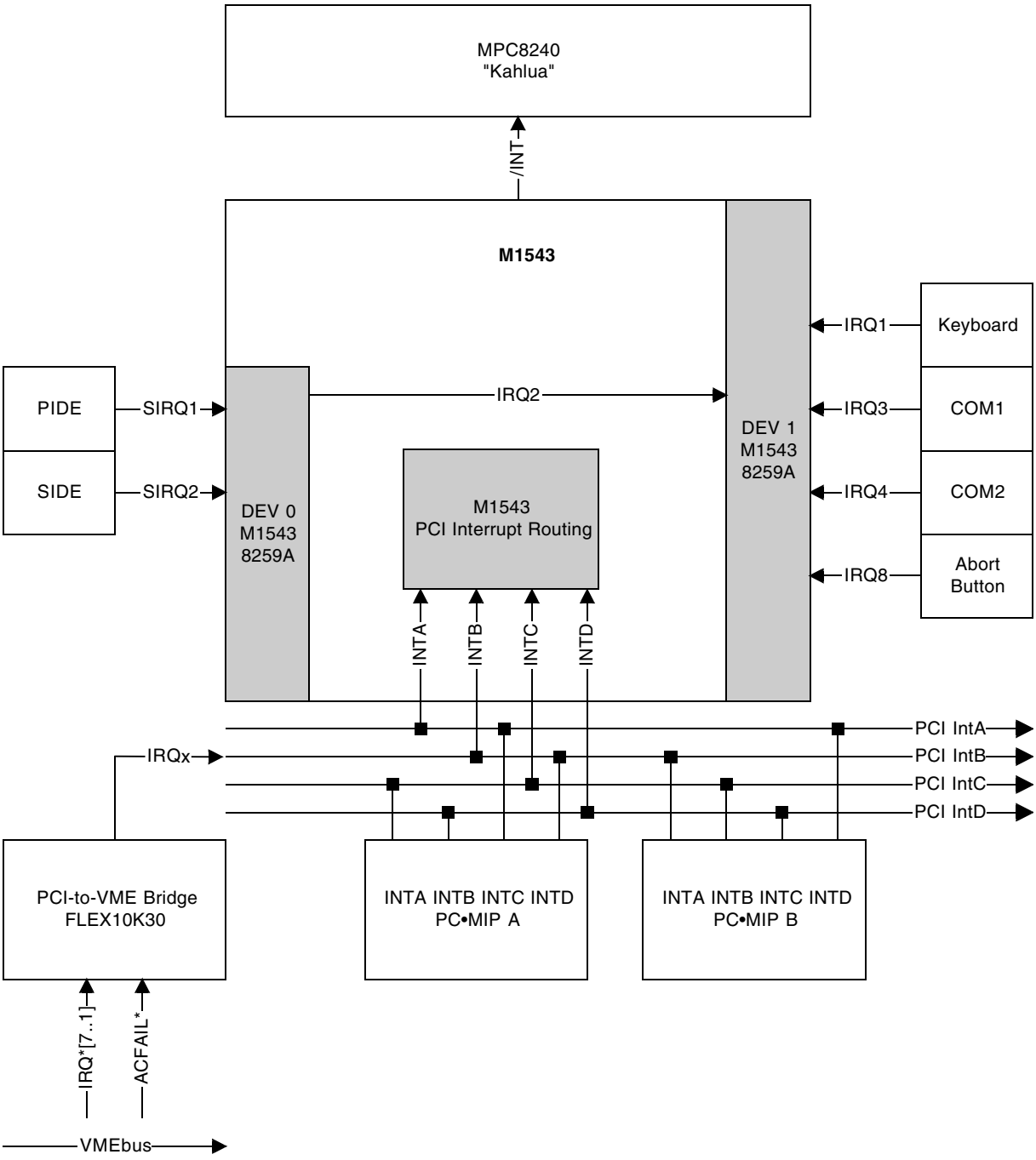


Table 24. *Interrupts on B11*

Interrupt	Active Polarity	Edge/Level	Source
0	high	Edge	Timer/Counter 0
1	high	Edge	Keyboard
3	high	Edge	COM2
4	high	Edge	COM1
7	low	Level	PCI INTA
8	low	Edge	ABORT
9	low	Level	PCI INTB (VME bridge)
10	low	Level	PCI INTC
11	low	Level	PCI INTD
12	high	Edge	Mouse
13			Not usable (Coprocessor INT in PC environment)
14	high	Edge	Primary IDE (CompactFlash) SIRQ1
15	high	Edge	Secondary IDE (Std IDE) SIRQ2

4.2.1 Nonmaskable Interrupts

The M1543 can be programmed to assert an NMI when it detects a low level of the SERR# line on the PCI local bus. The MPC107 will assert MCP# to the processor upon detecting a high level on NMI from the M1543. The MPC107 can also be programmed to assert MCP# under other conditions. Please refer to the MPC107 user manual for details.

4.2.2 Maskable Interrupts

The M1543 supports 15 interrupt requests. These 15 interrupts are ISA-type interrupts that are functionally equivalent to two 82C59 interrupt controllers. The chip also provides two steerable IRQ lines which can be routed to any of the available ISA interrupts. The M1543 supports four PCI interrupts: INTA#, INTB#, INTC# and INTD#. The interrupt lines may to be routed to any of twelve ISA interrupt lines.

The entire interrupt routing is managed by the boot software and board support package of the operating system.

Note: All interrupts are handled by the ALI1543C PIC. The MPC8245's EPIC is not used!

4.3 Implementation of M1543 PCI-to-ISA Bridge

The GPO/GPI/GPIO pins of the M1543 are used for several functions on the B11. The tables below show the port assignments of the B11.

Table 25. *M1543 General Purpose Input (GPI) Pin Assignments*

GPI	Description
0	Abort button
1	Hex switch bit 1. This pin cannot be read, see Chapter 5.2 Board Revisions on page 83.
2	Hex switch bit 2
3	Hex switch bit 3

Table 26. *M1543 General Purpose Input/Output (GPIO) Pin Assignments*

GPIO	Direction	Description
0	in	Hex switch bit 0
1	in/out	If programmed to output low, reset CPU
2	out	Watchdog trigger (toggle)
3	in/out	SMB2 SCL
4	out	LED1
5	out	LED2
6	out	Reserved
7	in/out	SMB2 SDA

4.4 SMB Devices

Two System Management Buses are used: SMB 1 is handled via the M1543 SMB controller, SMB 2 via the GPIOs of the M1543.

Table 27. *SMB 1 Devices*

Address	Function
0x A0	SPD of SODIMM
0x 9A	LM75
0x D0	RTC M41T56

Table 28. *SMB 2 Devices*

Address	Function
0x 9x	Config Regs of SMS24
0x Ax	Memory Array of SMS24

4.5 PCI Devices on Bus 0

Table 29. *PCI Devices on Bus 0*

Device Number	Vendor ID	Device ID	Function
0x 00	0x 1057	0x 0003	MPC107 in MPC8245
0x 0C	0x 10B9	0x 5237	M1543 USB
0x 12	0x 10B9	0x 1533	M1543 PCI-to-ISA
0x 19	0x 1172	0x 5056	MEN VME bridge
0x 1A	-	-	PC•MIP Slot A
0x 1B	0x 10B9	0x 5229	M1543 IDE
0x 1C	0x 10B9	0x 7101	M1543 PMU
0x 1D	-	-	PC•MIP Slot B

5 Appendix

5.1 Literature and WWW Resources

5.1.1 Bridges

- MPC107 Host Bridge:
  MPC107 PCI Bridge/Memory Controller User's Manual, Motorola
  www.mot.com
- M1543 PCI-to-ISA bridge:
  M1543 Preliminary Data Sheet, Acer Laboratories Inc. Jan. 1998 / Version 1.25
  www.acerlabs.com

5.1.2 PCI

- PCI Local Bus Specification Revision 2.1:
  1995; PCI Special Interest Group
  P.O. Box 14070
  Portland, OR 97214, USA
  www.pcisig.com

5.1.3 PC•MIP

- PC•MIP Standard:
  standard ANSI/VITA 29;
  VMEbus International Trade Association
  7825 E. Gelding Dr., Ste. 104,
  Scottsdale, AZ 85260
  www.vita.com

5.2 Board Revisions

Table 30. *Table of Hardware Revisions*

<table>
<tr><th>Revision</th><th>Comment</th><th>Restrictions</th></tr>
<tr><td>00.xx</td><td>Prototype</td><td>IEEE 1149.1 connector mirrored
Hex switch not populated
Only selected SDRAM types will work</td></tr>
<tr><td>01.xx</td><td>First revision</td><td rowspan="2">Hex switch is populated as of revision 01.02.xx. However, GPI1 cannot be read. Therefore, hex switch settings are read as follows:
<table>
<tr><th>Hex Switch</th><th>Read Value</th><th>Hex Switch</th><th>Read Value</th></tr>
<tr><td>0</td><td>0</td><td>8</td><td>8</td></tr>
<tr><td>1</td><td>1</td><td>9</td><td>9</td></tr>
<tr><td>2</td><td>0</td><td>A</td><td>8</td></tr>
<tr><td>3</td><td>1</td><td>B</td><td>9</td></tr>
<tr><td>4</td><td>4</td><td>C</td><td>C</td></tr>
<tr><td>5</td><td>5</td><td>D</td><td>D</td></tr>
<tr><td>6</td><td>4</td><td>E</td><td>C</td></tr>
<tr><td>7</td><td>7</td><td>F</td><td>D</td></tr>
</table></td></tr>
<tr><td>02.xx</td><td>Second revision</td></tr>
</table>

5.3 Component Plans

Figure 16. *Component Plan of B11 Rev. 02 - Top Side*

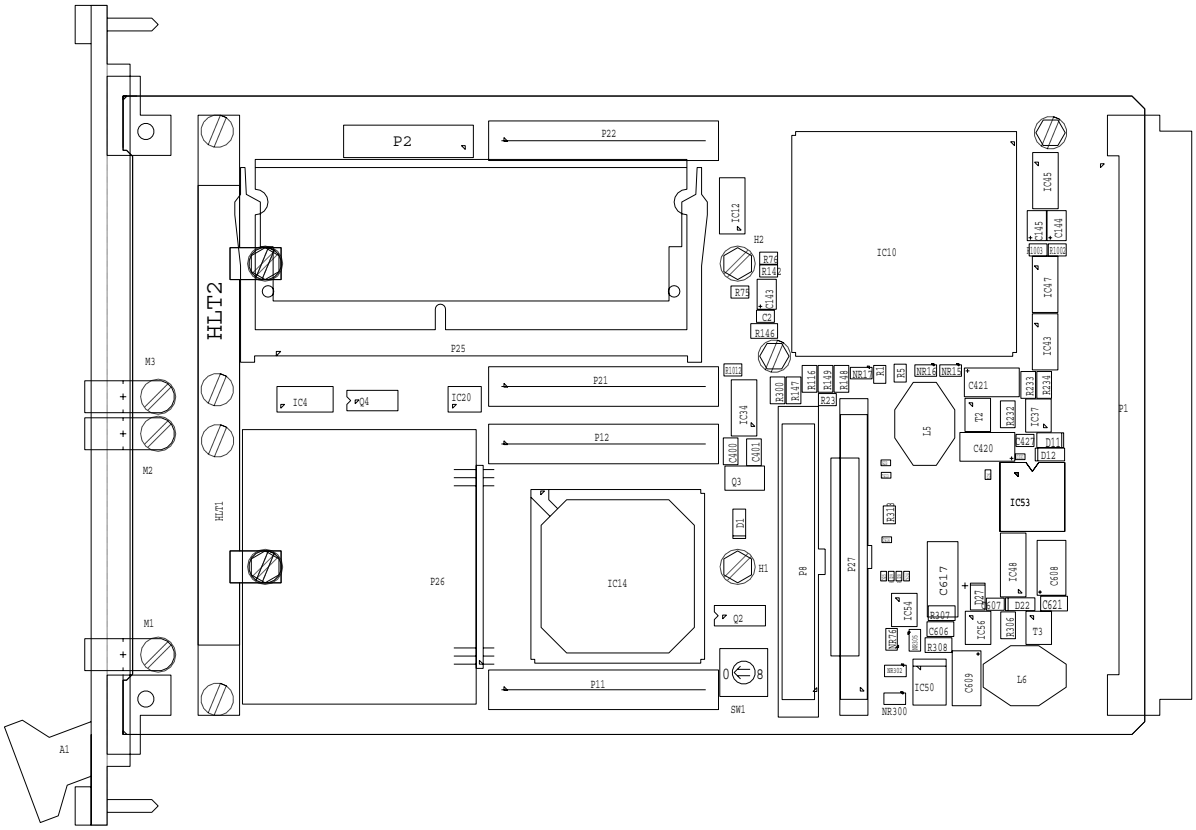
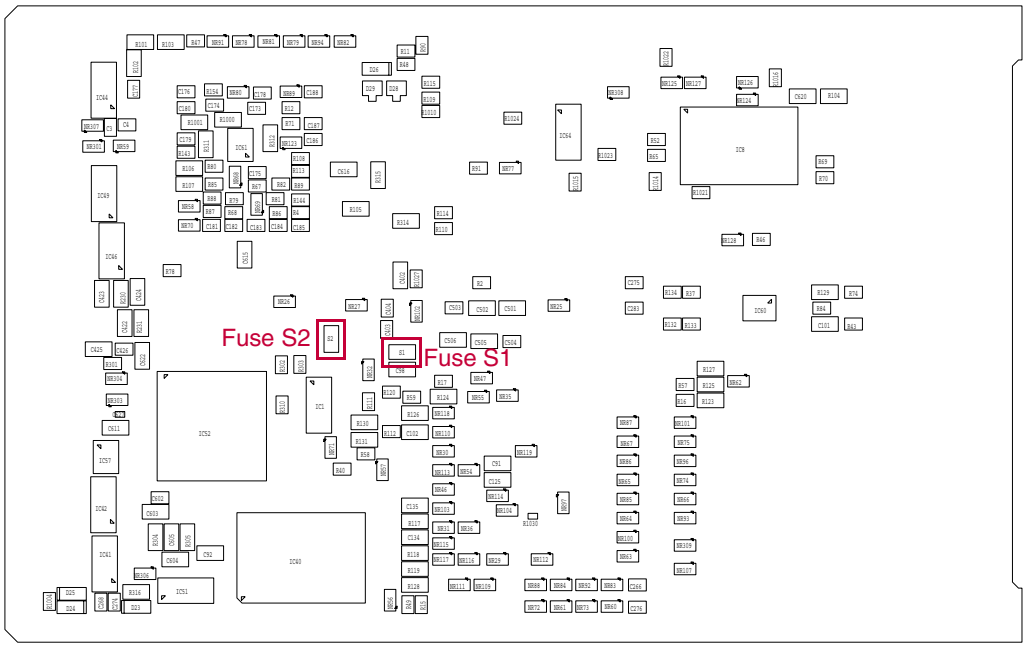


Figure 17. *Component Plan of B11 Rev. 02 - Bottom Side*



You can request the circuit diagrams for the current revision of the product described in this manual by completely filling out and signing the following non-disclosure agreement.

Please send the agreement to MEN by mail. We will send you the circuit diagrams along with a copy of the completely signed agreement by return mail.

MEN reserves the right to refuse sending of confidential information for any reason that MEN may consider substantial.

mikro elektronik
gmbh · nürnberg

Non-Disclosure Agreement

for Circuit Diagrams provided by MEN Mikro Elektronik GmbH

between

MEN Mikro Elektronik GmbH
Neuwieder Straße 7
D-90411 Nürnberg

("**MEN**")

and

_____
_____
_____
_____

("**Recipient**")

We confirm the following Agreement:

MEN		**Recipient**	
Date:	_____	Date:	_____
Name:	_____	Name:	_____
Function:	_____	Function:	_____

Signature: Signature:

_____ _____

The following Agreement is valid as of the date of MEN's signature.

MEN Mikro Elektronik GmbH

Neuwieder Straße 7
90411 Nürnberg
Deutschland

Tel. +49-911-99 33 5-0
Fax +49-911-99 33 5-99

E-Mail info@men.de
www.men.de

Geschäftsführer Manfred Schmitz, Udo Fuchs Handelsregister Nürnberg HRB 5540 UST-ID-Nr. DE 133 528 744

Deutsche Bank AG Kto. Nr. 0390 211, BLZ 760 700 12 HypoVereinsbank Kto. Nr. 1560 224 300, BLZ 760 200 70

ISO 9001 zertifiziert

1 Subject

The subject of this Agreement is to protect all information contained in the circuit diagrams of the following product:

Article Number: _____ [filled out by recipient]

MEN provides the recipient with the circuit diagrams requested through this Agreement only for information.

2 Responsibilities of MEN

Information in the circuit diagrams has been carefully checked and is believed to be accurate as of the date of release; however, no responsibility is assumed for inaccuracies. MEN will not be liable for any consequential or incidental damages arising from reliance on the accuracy of the circuit diagrams. The information contained therein is subject to change without notice.

3 Responsibilities of Recipient

The recipient, obtaining confidential information from MEN because of this Agreement, is obliged to protect this information.

The recipient will not pass on the circuit diagrams or parts thereof to third parties, neither to individuals nor to companies or other organizations, without the written permission by MEN. The circuit diagrams may only be passed to employees who need to know their content. The recipient protects the confidential information obtained through the circuit diagrams in the same way as he protects his own confidential information of the same kind.

4 Violation of Agreement

The recipient is liable for any damage arising from violation of one or several sections of this Agreement. MEN has a right to claim damages amounting to the damage caused, at least to €100,000.

5 Other Agreements

MEN reserves the right to pass on its circuit diagrams to other business relations to the extent permitted by the Agreement.

Neither MEN nor the recipient acquire licenses for the right of intellectual possession of the other party because of this Agreement.

This Agreement does not result in any obligation of the parties to purchase services or products from the other party.

6 Validity of Agreement

The period after which MEN agrees not to assert claims against the recipient with respect to the confidential information disclosed under this Agreement shall be _____ months [filled out by MEN]. (Not less than twenty-four (24) nor more than sixty (60) months.)

7 General

If any provision of this Agreement is held to be invalid, such decision shall not affect the validity of the remaining provisions and such provision shall be reformed to and only to the extent necessary to make it effective and legal.

This Agreement is only effective if signed by both parties.

Amendments to this Agreement can be adopted only in writing. There are no supplementary oral agreements.

This Agreement shall be governed by German Law.

The court of jurisdiction shall be Nuremberg.

MEN Mikro Elektronik GmbH

Neuwieder Straße 7
90411 Nürnberg
Deutschland

Tel. +49-911-99 33 5-0
Fax +49-911-99 33 5-99

E-Mail info@men.de
www.men.de

Geschäftsführer Manfred Schmitz, Udo Fuchs Handelsregister Nürnberg HRB 5540 UST-ID-Nr. DE 133 528 744

Deutsche Bank AG Kto. Nr. 0390 211, BLZ 760 700 12 HypoVereinsbank Kto. Nr. 1560 224 300, BLZ 760 200 70

ISO 9001 zertifiziert